

# Utbedring og uttesting av programmet fap2D

**Torjus Sandviken**  
**Espen Shin Skogsrud**

Master i ingeniørvitenskap og IKT  
Innlevert: juni 2014  
Hovedveileder: Kjell Magne Mathisen, KT  
Medveileder: Kolbein Bell, KT

Norges teknisk-naturvitenskapelige universitet  
Institutt for konstruksjonsteknikk





## MASTEROPPGAVE 2014

FAGOMRÅDE: Konstruksjonsteknikk / Programutvikling	DATO: 10. juni 2014	ANTALL SIDER: 14 + 84 + 214
--	------------------------	--------------------------------

TITTEL:

**Utbedring og uttesting av programmet fap2D**

Improving and validating the program **fap2D**

UTFØRT AV:

Torjus Sandviken  
Espen Shin Kraavik Skogsrud



SAMMENDRAG:

Denne rapporten presenterer arbeidet med å videreutvikle programmet **fap2D**, et PC-program for statisk og dynamisk analyse av plane rammekonstruksjoner.

Arbeidet bygger videre på arbeider utført i tidligere masteroppgaver knyttet til **fap2D**. Til sammen har det blitt utviklet et omfattende analyseverktøy for konstruksjoner. Etter hvert som programmet har vokst har det blitt stadig mer utfordrende å holde det feilfritt og brukervennlig. Hovedfokuset i denne masteroppgaven har derfor vært å rette opp feil og gjøre brukeropplevelsen bedre.

Rapporten viser en oversikt over endringer som har blitt gjort i programmet siden forrige versjon. Endringene er i form av feilrettinger, mindre justeringer av funksjonalitet og nye implementasjoner. Deler av endringene er basert på tilstandsrapporter som er skrevet i tidligere arbeider, blant annet de undertegnede prosjektoppgaver fra høsten 2013. Likevel er størstedelen basert på vurderinger gjort undervegs i masteroppgaven.

I tillegg til selve programmeringsarbeidet er det gjennomført drøftinger av tilstanden og fremtiden til **fap2D**. Drøftingene er samlet i en rapport som tar for seg temaer som betatesting, lisensiering og distribusjon, validering av analyser, og gjenstående programmeringsoppgaver. For å fremskynde læringsprosessen for fremtidige utviklere har man forsøkt å etterlate seg detaljert dokumentasjon av kompliserte deler av kildekoden.

Hovedproduktet av masteroppgaven er programmet **fap2D** i seg selv. For å gi et mer detaljert inntrykk av hva som er gjort er det lagt til flere større vedlegg som beskriver arbeidet. Kronologisk logg for begge de undertegnede viser fremdriften.

HOVEDVEILEDER:	Kjell Magne Mathisen
MEDVEILEDER:	Kolbein Bell
UTFØRT VED:	Institutt for konstruksjonsteknikk, NTNU.



# Summary

This report presents the effort to further develop the application **fap2D**, a PC program for static and dynamic analysis of 2D frame type structures.

The work builds on works carried out in earlier theses related to **fap2D**. All together, a comprehensive tool for structural analysis has been developed. As the program has expanded in size it has become increasingly challenging to keep it free of errors and user friendly. The main focus in this thesis has therefore been to correct errors and improve the user experience.

The report lists all the changes made to the program since the previous version. The changes consist of bug fixes, smaller modifications of functionality and completely new implementations. Some of the alterations are based on condition reports written in previous works, yet the majority is based on assessments during the course of the thesis.

Apart from the actual programming work, a chapter has been dedicated to discuss the state and future of **fap2D**. The discussions are gathered in a state report that addresses topics such as beta testing, licensing and distribution, validation of analyzes, and remaining tasks. Important parts of the source code has been documented in detail as an attempt to accelerate the learning process for future developers.

The main product of the thesis is the application **fap2D** itself. In order to provide a more detailed view of what has been done, several major appendices describe the content in the report and the application. Chronological logs for the writers show the work progress.



## MASTEROPPGAVE 2014

for

*Torjus Sandviken og Espen Shin Kraavik Skogsrud*

### Utbedring og uttesting av programmet **fap2D**

#### *Improving and validating program **fap2D***

Basert på utført prosjektoppgave høsten 2013 går oppgaven ut på å videreføre utviklingen av Windows-programmet **fap2D**, et program for statiske og dynamiske beregninger av plane ramme-konstruksjoner som ble påbegynt allerede i 2006. Frem til våren 2013 har 11 studenter deltatt i utviklingen av programmet, som prosjekt- og masteroppgaver.

Fokus i denne oppgaven er programmets kvalitet, ikke nye beregningsmetoder, og målet er et korrekt, stabilt og robust program. Stikkord i denne sammenheng er: feilretting, uttesting, validering, lagring, utskrift/eksport, «settings» («customize/options») og hjelpe-funksjon via «tooltip» og eventuelt «tutorial».

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

Besvarelsen er først og fremst selve programmet, men det skal også utarbeides en 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. Rapporten bør også omfatte informasjon, gjerne som vedlegg, som kan være nyttig/nødvendig for andre som skal sette seg inn i koden og eventuelt vedlikeholde/modifisere/videreføre den.


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

*Hovedveileder:* Kjell Magne Mathisen

*Medveileder:* Kolbein Bell

**Besvarelsen skal leveres til Institutt for konstruksjonsteknikk innen 10. juni 2014.**

NTNU, 18. januar, 2013

  
Kjell Magne Mathisen





# Forord

I 2006 startet professor emeritus Kolbein Bell og studentene Sverre Eide Holst og Magnus Minsaas arbeidet med å lage et PC-program for analyse av plane rammekonstruksjoner. Programmet fikk arbeidstittelen **Frame Analysis Program 2D**, bedre kjent som **fap2D**.

Siden 2006 har programmet blitt kontinuerlig utvidet med nye analysetyper. Bell har vært ansvarlig for beregningsmetodene, kalt **Frame2D**, mens studentene har laget brukergrensesnittet for å modellere konstruksjoner og presentere resultater av beregninger.

Til sammen har tretten studenter ved Institutt for konstruksjonsteknikk bidratt i utviklingen. Da Frans Erstad, Kristian Pedersen og Erik Aasmundrud avsluttet sine masteroppgaver våren 2013 gav de stafettpinnen videre til oss. Høsten 2013 skrev vi parallelt hver vår prosjektoppgave der vi gjorde en tilstandsvurdering av **fap2D** og kartla endringer som burde implementeres i programmet. Tilstandsvurderingen viste at programmet i sin helhet fungerer hensiktsmessig, men at det bør gjøres mer stabilt, robust og brukervennlig.

Vi vil rette en stor takk til primus motor Kolbein Bell for entusiastisk veiledning gjennom det siste snaue året. Bell har vært tilgjengelig til en hver anledning og bidratt med idéer, gode diskusjoner, testing og vedlikehold av beregningskjerne og brukermanual. Vi vil også takke hverandre for godt samarbeid i den samme perioden. Vi har hatt en god dialog og fordelt arbeidsoppgavene broderlig.


Faglærer Arne Aalberg og studenter i faget TKT4180 KMEK - Beregningsmetoder takkes for å ha gjort det mulig å gjennomføre betatesting. Vit.ass. i samme fag, Petter Henrik Holmstrøm, takkes for gode innspill til forbedringer og ny funksjonalitet. Dessuten skal overingeniør Kenneth Sundli ha takk for hjelp til installasjon av **fap2D** på datasaler i forbindelse med betatesting. Ellers vil vi takke våre medstudenter Henrik Thorsen, Erlend Orthe og Torger Nordgård for gode diskusjoner på kontoret og for at de hjalp til med å teste **fap2D** på Windows 8-plattformen, slik at vi kunne avgjøre om en betatesting lot seg gjennomføre.

Sist, men ikke minst skal alle de tidligere utviklerne ha honnør for innsatsen de har lagt ned, spesielt Aasmundrud, Pedersen, Erstad og Eide Holst som har bistått oss direkte.

Trondheim, juni 2014



Torjus Sandviken



Espen Shin Kraavik Skogsrud



# Innhold

<b>Summary</b>	<b>I</b>
<b>Oppgavetekst</b>	<b>III</b>
<b>Forord</b>	<b>V</b>
<b>Innhold</b>	<b>VII</b>
<b>Akronymer</b>	<b>XI</b>
<b>1 Innledning</b>	<b>1</b>
1.1 Presisering og avgrensing av oppgaven . . . . .	2
1.2 Arbeidsprosessen . . . . .	3
1.2.1 Planlegging . . . . .	3
1.2.2 Forstudium . . . . .	3
1.2.3 Gjennomføring . . . . .	3
1.3 Struktur på rapporten . . . . .	6
<b>2 Definisjoner og terminologi</b>	<b>7</b>
2.1 Rapporten . . . . .	7
2.1.1 Formateringer . . . . .	7
2.1.2 Engelske uttrykk . . . . .	7
2.2 Programmet . . . . .	12
2.2.1 Brukergrensesnitt - GUI . . . . .	12
2.2.2 Konstruksjonsmodell og laster . . . . .	15
2.2.3 Beregningsmodell . . . . .	16
2.2.4 Analysetyper og resultater . . . . .	18
<b>3 Endringer i fap2D</b>	<b>19</b>
3.1 GUI . . . . .	19
3.1.1 Design . . . . .	19
3.1.2 Oppførselen til <i>controls</i> . . . . .	22
3.1.3 DevExpress-oppdateringer . . . . .	24
3.2 Tegning - OpenGL . . . . .	24
3.2.1 N- og V-diagrammer . . . . .	24
3.2.2 Mindre rettinger og modifiseringer av tegnetoder . . . . .	26
3.3 Markering av <i>structural items</i> . . . . .	28
3.3.1 Markering av <i>members</i> og <i>submembers</i> . . . . .	28
3.3.2 Markeringspreferanse . . . . .	30
3.3.3 Markering og visning av elementer . . . . .	31

3.3.4	Feilaktig <i>snap</i> ved museklikk . . . . .	32
3.3.5	Feilaktig markering av objekter vha. høyretasten . . . . .	33
3.4	Konstruksjonsmodell . . . . .	33
3.4.1	Mindre endringer i konstruksjonsmodell . . . . .	34
3.4.2	<i>Member</i> -retning . . . . .	34
3.4.3	Overlappende <i>structural items</i> . . . . .	36
3.4.4	<i>Arch members</i> . . . . .	37
3.4.5	Eksentrisiteter . . . . .	38
3.5	Beregningsmodell og analyser . . . . .	40
3.5.1	Standardverdier for <i>load combinations</i> og <i>time axes</i> . . . . .	40
3.5.2	Konvertering av laststeg til statisk lasttilfelle . . . . .	41
3.5.3	Unøyaktig beregning av nodeplassering langs <i>arch members</i> . . . . .	41
3.5.4	Nedre grense for elementlengde . . . . .	42
3.6	Presentasjon av resultater . . . . .	45
3.6.1	Strekk- og trykkrefter . . . . .	45
3.6.2	Markering av maksimalrespons i resultatdiagrammer . . . . .	45
3.6.3	Maksimal forskyvningsvektor . . . . .	45
3.6.4	Forbedret visning av spenninger . . . . .	46
3.6.5	Presentasjon og navigering av elementresultater . . . . .	46
3.7	<i>Steel design</i> . . . . .	47
3.8	Eksport og import . . . . .	51
3.9	<i>Settings</i> . . . . .	53
3.9.1	Redigering og fjerning av eksisterende <i>settings</i> . . . . .	54
3.9.2	Lagring av <i>settings</i> . . . . .	54
3.9.3	Nye innstillinger . . . . .	54
3.10	Validering av input . . . . .	56
3.11	Informasjon til brukeren . . . . .	57
3.11.1	<i>Tooltips</i> . . . . .	57
3.11.2	Musepeker-koordinater i <i>status bar</i> . . . . .	58
3.11.3	Formatering av <i>error</i> og <i>information lists</i> . . . . .	58
3.11.4	Tusen-skilletegn og prosentvisning i tallverdier . . . . .	59
3.12	Filbehandling . . . . .	59
<b>4</b>	<b>Tilstandsrapport</b> . . . . .	<b>61</b>
4.1	Betatesting av <b>fap2D 3.0.0 Beta</b> . . . . .	61
4.2	Distribusjon, lisensiering og installasjon . . . . .	62
4.3	“3D-effekter” i <b>fap2D</b> . . . . .	62
4.3.1	Overlappende <i>members</i> . . . . .	62
4.3.2	Overlappende <i>joints</i> . . . . .	64
4.4	Videreføring av kunnskap . . . . .	64
4.5	Testing og validering . . . . .	65
4.5.1	Analyser . . . . .	66
4.5.2	Validering av <i>arch members</i> . . . . .	66
4.5.3	Parabelkoordinater . . . . .	67
4.5.4	Validering av eksentrisiteter . . . . .	68
4.6	Dokumentasjon for utviklere . . . . .	71
4.6.1	Kommentarer i kildekode . . . . .	71
4.6.2	Fargebruk i <b>fap2D</b> . . . . .	72

4.6.3	<i>Mouse Events</i> . . . . .	72
4.6.4	<i>GridView Events</i> . . . . .	72
4.6.5	Reviderte <i>tooltips</i> . . . . .	73
4.6.6	Installasjon og konfigurasjon av programvare . . . . .	73
4.7	Gjenstående oppgaver . . . . .	73
4.7.1	Oppgaver som bør gjennomføres . . . . .	74
4.7.2	Oppgaver som kan ses bort fra . . . . .	79
4.7.3	Feil som ikke er gjenskt . . . . .	80
<b>5</b>	<b>Konklusjon og videre arbeid</b>	<b>81</b>
5.1	Konklusjon . . . . .	81
5.2	Anbefalinger for videre arbeid . . . . .	82
	<b>Referanser</b>	<b>83</b>
<b>A</b>	<b>Release notes - Versjonsmerknader</b>	<b>1</b>
	Introduksjon . . . . .	1
A.1	GUI . . . . .	2
A.2	Tegning - OpenGL . . . . .	25
A.3	Markering av <i>structural items</i> . . . . .	35
A.4	Konstruksjonsmodell . . . . .	41
A.5	Beregningsmodell og analyser . . . . .	75
A.6	Resultater . . . . .	85
A.7	<i>Steel design</i> . . . . .	96
A.8	Eksport og import . . . . .	101
A.9	<i>Settings</i> . . . . .	106
A.10	Validering av input . . . . .	118
A.11	Informasjon til brukeren . . . . .	121
A.12	Filbehandling . . . . .	127
<b>B</b>	<b>Betatesting av fap2D 3.0.0</b>	<b>133</b>
B.1	Introduksjon til betatesting . . . . .	133
B.2	Gjennomføring av betatesting . . . . .	133
<b>C</b>	<b>Dokumentasjon for utviklere</b>	<b>144</b>
C.1	Fargebruk i <b>fap2D</b> . . . . .	144
C.2	Reviderte <i>tooltips</i> . . . . .	148
C.3	<i>GridView events</i> . . . . .	172
C.4	<i>Mouse events</i> . . . . .	175
C.5	Testing av analyser ( <i>tutorials</i> ) . . . . .	182
<b>D</b>	<b>Logg</b>	<b>186</b>
D.1	Logg - Torjus Sandviken . . . . .	186
D.2	Logg - Espen Skogsrud . . . . .	194
<b>E</b>	<b>Gantt-diagram</b>	<b>201</b>
<b>F</b>	<b>Programvare</b>	<b>202</b>
F.1	Systemkrav for fap2D . . . . .	202
F.2	Programvare brukt i utviklingen . . . . .	203

<b>G</b>	<b>Forklaring av kildekoden</b>	<b>204</b>
G.1	Navnekonvensjoner . . . . .	204
G.2	Prosjektoversikt . . . . .	205
G.3	Datasekk med <i>global settings</i> . . . . .	211
G.4	Nøkkeltall - <i>Code metrics</i> . . . . .	212
<b>H</b>	<b>Innhold på vedlagt minnepenn</b>	<b>214</b>
H.1	Rapport . . . . .	214
H.2	Installasjonsfiler . . . . .	214
H.3	Kildekode . . . . .	214
H.4	Referanser . . . . .	214

# Akronymer

Akronymer som er brukt i rapporten.

<b>fap2D</b>	Frame Analysis Program 2D
<b>Frame2D</b>	Beregningskjernen i <b>fap2D</b>
<b>3.0</b>	Versjon <b>3.0.0 Beta</b> av <b>fap2D</b> - forrige versjon
<b>3.1</b>	Versjon <b>3.1</b> av <b>fap2D</b> - nyeste versjon
DX	DevExpress - rammeverk for å lage GUI
GUI	Grafisk brukergrensesnitt
KT	Institutt for konstruksjonsteknikk, NTNU
SVN	<i>Subversion</i> - versjonskontroll
UM	<i>User's Manual</i> - brukermanual
WF	Windows Forms - Et basisrammeverk for å lage GUI (som DX er basert på)





# Kapittel 1

## Innledning

Det er mange aktører innen utvikling av programvare for analyse av konstruksjoner.<sup>1</sup> Programvare har ingen landegrenser og dersom man skal synes i mengden er det viktig å være klar over sine fortrinn og bruke dem til å treffe en målgruppe.



Figur 1.1: Oppstartsbilde for den nye versjonen, **fap2D 3.1**

At utviklingen av **fap2D** ikke har vært drevet av kommersielle målsetninger gjør at man har kunnet utvikle en programvare som skiller seg fra en del andre.

Der kommersielle aktører fokuserer på innfløkt (og derfor innbringende) dimensjonering av konstruksjoner i hht. standarder, fokuserer **fap2D** på grunnleggende statiske og dynamiske analyser som kan kontrolleres vha. enkel håndregning. Der kommersiell programvare gjerne modellerer konstruksjoner vha. to- og tredimensjonale elementer, konsentrerer **fap2D** seg om modellering av enkle rammekonstruksjoner i planet. Rutenettet i det todimensjonale modellvinduet i programmet gir tydelige assosiasjoner til sidene i ingeniørstudenters kladdebøker, fylt med tegninger av de samme konstruksjonene.

Målgruppen for **fap2D** bør derfor være studenter og andre som er tidlig i lærefasen og motiveres av å benytte et enkelt og intuitivt program. Filosofien i **fap2D** er nemlig at man raskt skal kunne konstruere og analysere modeller vha. et enkelt og intuitivt grensesnitt. Det gjør at **fap2D** er et passende steg på studentenes vei til å beherske mer avansert programvare.

<sup>1</sup>Vi nevner i farten noen programmer: Focus Konstruksjon [12], Abaqus [19] og Tekla [16].

Intensjonen i mye av arbeidet som er lagt ned i denne rapporten er å tilpasse programmet ytterligere til målgruppen. Det er gjort ved å gi brukeren mer informasjon og rettleiding, og ved å forbedre modelleringsprosessen og resultatvisningene. Selv om mange av endringene skal komme målgruppen til gode mener man også de vil gagne brukerne generelt. Det er blant annet lagt til mulighet for eksport og import av data, og nye innstillinger som gir brukeren mer frihet.

## 1.1 Presisering og avgrensing av oppgaven

Oppgaveteksten beskriver hensikten med masteroppgaven, men den presiseres i dette avsnittet. Tittelen “Utbedring og uttesting av programmet fap2D” beskriver det overordnede målet i oppgaven. For å utbedre programmet har vi valgt å fokusere på utbedringer som er viktige for brukerne, spesielt målgruppen, studenter.

Feil som er synlige i GUI og feil i konstruksjons- og beregningsmodellen er høyt prioritert. Det samme er nye funksjoner som tilfører nytte. Tabell 1.1 nedenfor viser de konkrete delmålene ved masteroppgavens start.

Tabell 1.1: Delmål.

### Korrekthet

---

1. Rette feil i GUI.
2. Rette feil i konstruksjonsmodell.
3. Melde fra til veileder om evt. feil i beregningsmodell, slik at de kan rettes.
4. Teste GUI, konstruksjonsmodell (og beregningsmodell i den grad det er tid).

### Stabilitet og robusthet

---

1. Verifisere om installasjon og programkjøring fungerer for ulike plattformer.
2. Gjøre lagring og lasting av filer, og annen filbehandling mer robust.
3. Rette feil som forårsaker programstopp, såkalte *unhandled exceptions*.
4. Innføre strengere validering av input fra brukeren.
5. Gjøre modelleringsprosessen mer robust. Prosessen tar ikke høyde for at konstruksjoner kan endres undervegs, den antar at alt gjøres “korrekt” i første omgang.

### Endringer og ny funksjonalitet

---

1. Eksport av modell og resultater til tekstfil.
2. Import av modellgeometri (i form av *joint coordinates*).
3. Revidering av *tooltips*, *settings* og innholdet i *User's Manual*. Veileder er ansvarlig for UM.
4. Hjelpesfunksjonalitet i form av *tutorial*.
5. Generelle forbedringer: Tegning, modellering, resultatvisning og brukerinteraksjon.

Optimalisering av ytelsen og ressursbruken til **fap2D** er ikke prioritert. Programmet fungerer tilfredsstillende for de fleste PC-er av nyere dato. Dette er i tråd med vedlegg F.1 om systemkrav. Det understrekes at oppgaven først og fremst består av kvalitative vurderinger, ikke kvantitative.

## 1.2 Arbeidsprosessen

### 1.2.1 Planlegging

Prosjektoppgavene, høsten 2013[10] [11], ble brukt til å kartlegge endringer som burde utføres og til å vurdere den overordnede tilstanden til programmet. Det ble også satt av tid til å lære kildekoden å kjenne, som en forberedelse til masteroppgaven. Enkel validering av statiske og dynamiske analyser ble gjort, men ikke omfattende.

Det ble satt opp en prioriteringsliste for feil og endringer og laget en grov fremdriftsplan for masteroppgaven.<sup>2</sup> Punktene i prioriteringslisten er vurdert mhp. nytteverdi, anslått tidsforbruk og krav til forkunnskaper. Dvs. at nytteverdien er viktigst, tidsbruken nest viktigst og kravet til forkunnskaper minst viktig.

Fremdriftsplanen ble laget av Sandviken, før det ble avgjort om masteroppgaven skulle skrives i fellesskap eller individuelt. Planen ble brukt som et utgangspunkt for den felles masteroppgaven.

Planleggingen av masteroppgaven ble gjennomført i januar 2014. Den bygde videre på den nevnte tidsplanen fra prosjektoppgaven. Perioden satt av til validering av analyser ble forskjøvet til slutten av oppgaven.

### 1.2.2 Forstudium

Både i prosjekt- og masteroppgaven er det brukt tid på å studere tidligere masteroppgaver relatert til **fap2D**; Spesielt oppgavene til Pedersen [9], Erstad [8] og Aasmundrud [6], siden de er de nyligste oppgavene. Pedersens oppgave har flest fellestrekk med denne masteroppgaven.

I tillegg har det blitt brukt mye tid på å sette seg inn i den eksisterende kildekoden. Dessverre er koden tynt kommentert enkelte steder, noe som har gjort forstudiet ekstra omfattende. Kildekoden er også basert på mange ulike rammeverker som man må bruke tid på å sette seg inn i.

### 1.2.3 Gjennomføring

#### 1.2.3.1 Strategi

Etter planleggingsfasen, startet vi masteroppgaven med å rette noen åpenbare feil i **fap2D**. Blant annet knyttet til *arch members* og tegning av aksial- og skjærkraftdiagrammer. Det viste seg at deler av oppgavene var utfordrende på dette tidspunktet. Derfor endret vi i februar fokus til enklere oppgaver for å bli varme i trøya. Undervegs har fokuset skiftet flere ganger. Grunnen er at listen over oppgaver til tider har vært lang. En porsjon av de mindre oppgavene har blitt utført for å korte ned listen. Deretter har man gått løs på de tyngre oppgavene igjen, i hht. avsnitt 1.2.1.

---

<sup>2</sup>Fremdriftsplanen er gjengitt i vedlegg E.1.

Det er lagt omtrent like stor vekt på å dokumentere som det er lagt vekt på å implementere. Dette fordi det er viktig å være klar over svakhetene i programmet. Ved bare å fokusere på løse problemer etter hvert som man oppdager dem, overser man lett det man ikke har gjort.

For å tegne et ærlig bilde av tilstanden nevnes mange oppgaver som ikke er blitt utført, men kanskje burde vært det.<sup>3</sup>

### 1.2.3.2 Utviklingsarbeidet

Tre personer har deltatt i denne fasen av utviklingen av **fap2D**, vi to samt veileder. Vi har jobbet med grensesnittet, som er skrevet i språket *C#* vha. utviklingsmiljøet Visual Studio[13]. For å kunne arbeide med de samme filene uten å skape skrive/lese-konflikter har vi benyttet en versjonskontroll vha. TortoiseSVN [23], VisualSVN [24] og PuTTY [22]. Versjonskontrollen har også sørget for å holde oversikt over alle endringene, slik at vi kunne tilbake stille om nødvendig. Prosedyren som ble brukt for å sette opp versjonskontroll er gjengitt i [6, vedlegg F. SVN].

En fullstendig oversikt over plattformen som er brukt finnes i vedlegg F.2.



Vi har samarbeidet om å fordele arbeidsoppgaver, og vi har ofte arbeidet uavhengig med å løse spesifikke oppgaver, men har løpende gitt hverandre innspill og hjelp. De vanskeligste oppgavene er ofte løst i fellesskap.

Veileder Bell har hatt ansvar for koden i beregningskjernen og for *User's Manual*(UM). Vi har kommet med innspill til endringer som veileder så har gjennomført. Dette har fungert godt.

Erstad, Pedersen og Aasmundrud formulerte våren 2013 et dokument med forslag til kodekonvensjoner [6, vedlegg E.1] som vi har forholdt oss til i vårt programmeringsarbeid.

### 1.2.3.3 Planlegging og møter

Teamet har hatt ukentlige møter der man har diskutert arbeidet som ble gjort uken før, og planlagt arbeidet for den kommende uken. Møtene har vært viktige for å gjøre opp status underveis og peile ut veien videre. I praksis har de vært mye viktigere enn fremdriftsplanen vi først la.

I tillegg til møtene har styringsdokumentet vært en liste med prioritering av feilretting og ny funksjonalitet.

<sup>3</sup>De gjenstående oppgavene finnes i kapittel 4.7.

### 1.2.3.4 Rapportering og dokumentasjon

Alle endringene som er gjort har blitt dokumentert i vedlegget A *Release notes* - Versjonsmerknader. Det viser hvem som har rapportert og hvem som har implementert de enkelte endringene.

Det har blitt ført logg over arbeidet som har blitt gjort hver dag, for eget bruk. Loggene som presenteres i vedlegg B.2.6 er oppsummeringer av arbeidet som er gjort i løpet av hver uke, og går ikke inn på detaljer. Vi ser ikke behovet for å presentere en omfattende logg, siden vedlegg A viser og argumenterer for alt som er gjort.

### 1.2.3.5 Begrensninger

Det er noen begrensninger i prosjektet **fap2D** som har forsinket fremdriften i arbeidet.

Tabell 1.2: Begrensninger.

#### Manglende videreføring av kunnskap

---

1. Manglende dokumentasjon av kode og manglende kommentering i koden.
2. Manglende argumentasjon for arbeider gjort i tidligere masteroppgaver.
3. Manglende tilgang på masteroppgavene til de fire første utviklerne.

#### Utviklingsarbeidet

---

1. Ikke benyttet integrert *bug tracking system*.<sup>4</sup>
2. Manglende erfaring med programutvikling i team.
3. Bruk av delvis utdaterte rammeverk, som ikke vedlikeholdes [21].
4. Manglende erfaring med bruk av ulike rammeverk.

### 1.2.3.6 Endringer

Underveis i oppgaven har prioriteringen av oppgaver blitt endret. Feks. har det blitt brukt mer tid på retting av feil og implementasjon av ny funksjonalitet, enn på validering av analyser. Omfanget av de førstnevnte oppgavene ble muligens undervurdert, selv om det hele tiden har vært klart at det ville bli lite tid til analyser.

Det har blitt antatt at tidligere utviklere gjorde et grundig arbeid med å validere sine implementasjoner av analyser.

I hht. til fremdriftsplanen i vedlegg E.1 skulle det være et klart skille mellom perioden der vi rettet feil og perioden der vi implementerte ny funksjonalitet. Mye av kildekoden flyter sammen og det ble meningsløst med en slik inndeling. Feilsøking er tidkrevende og nye implementasjoner har ofte blitt prioritert høyere enn feilretting siden de er raskere å utføre, og tilfører nytteverdi.

Møtevirksomheten vår har styrt arbeidsprosessen, siden det har blitt noen avvik fra fremdriftsplanen.

## 1.3 Struktur på rapporten

Kapittel 2 beskriver definisjoner og terminologi som er brukt i resten av kapitlene.

Kapittel 3 gir en konkret innføring i endringene som er gjort i den nye versjonen av **fap2D**, versjon **3.1**. Både feilrettinger, justeringer og ny funksjonalitet.

Tilstandsrapporten i kapittel 4 tar for seg betatesting, gjenstående oppgaver, drøftinger og dokumentasjon som er laget for fremtidige utviklere. Temaer i drøftingene er lisensiering og distribusjon, testing av analyser, og kunnskapsoverføring i prosjektet **fap2D**.

Kapittel 5 oppsummerer rapporten og gir anbefalinger til videre arbeid i prosjektet.

I vedleggene bakerst finnes ytterligere dokumentasjon av endringer, i form av *release notes* - versjonsmerknader. Dette er det viktigste dokumentet i vedlegget. Ellers finnes en kort logg, systemkrav, forklaring av kildekoden, og dokumenter som utdyper deler av kapittel 4.

# Kapittel 2

## Definisjoner og terminologi

Dette kapitlet gir oversikt over formateringer og definisjoner som er brukt i rapporten og i **fap2D**. I tillegg gis det en innføring i programmets bestanddeler.

### 2.1 Rapporten

#### 2.1.1 Formateringer

Tabell 2.1 viser en oversikt over formateringene som er brukt i rapporten. Den er basert på en tilsvarende tabell fra avsnitt 2.1 i [3].

Tabell 2.1: Formateringer brukt i rapporten. [3]

Formatering	Beskrivelse
Normal	Vanlig tekst. Alle andre formateringer brukes i spesialtilfellene som er listet opp nedenfor.
<i>Kursiv</i>	Engelske ord og uttrykk som ikke har en passende oversettelse. Dette omfatter først og fremst uttrykkene som brukes i brukergrensesnittet til <b>fap2D</b> .
<b>Uthevet</b>	Brukes for særnnavnet <b>fap2D</b> , og i noen tilfeller til å fremheve viktig informasjon.
Maskinskrift	Brukes for kildekoden hentet fra <b>fap2D</b> .
Figur 1, tabell 1	Referanser til kapitler, figurer og tabeller i rapporten og i kilder og vedlegg.
[1]	Referanser til kilder.
<i>“Error message”</i>	Feilmeldinger fra programmet er skrevet i kursiv og med hermetegn.

#### 2.1.2 Engelske uttrykk

Siden brukergrensesnittet og kildekoden i **fap2D** er skrevet på engelsk er det naturlig å referere til engelske uttrykk i rapporten. Dette er gjort fordi mange av uttrykkene ikke har gode norske oversettelser. Dessuten mener vi det er en fordel at rapporten er mest mulig trofast mot terminologiene i programmet. Selv om det gjør språket litt kunstig blir det forhåpentligvis enklere for leseren å forstå innholdet. Tabell 2.2, 2.3, 2.4 og 2.5 forklarer

de viktigste engelske uttrykkene. Engelske uttrykk refereres alltid til i *kursiv*, derfor kan tabellene brukes som oppslagsverk når man har behov, man trenger ikke å lese dem i sin helhet.

Tabell 2.2: Engelske uttrykk som beskriver programmeringsuttrykk.

### Programmeringsterminologi

Engelsk	Norsk
<i>exception</i>	Et unntak, en mekanisme som fanger opp en oppførsel i koden som ikke er korrekt. Eksempelvis at man bruker tekst-input i regneoperasjoner.
<i>unhandled exception</i>	Et unntak som ikke er håndtert og fører til programstopp. Er ikke ønskelig.
<i>release version</i>	Ordinær (endelig) versjon av programvare.
<i>developer version</i>	Intern versjon, som bare benyttes undervegs i utviklingen for testing.
<i>beta version</i>	Siste versjon før ordinær utgivelse. Involverer brukerne for å få tilbakemeldinger.
<i>degugging</i>	Prosessen med å lete etter feil i kildekoden.

Tabell 2.3: Engelske uttrykk som beskriver brukergrensesnitt.

### Brukergrensesnitt

Engelsk	Norsk
GUI	Grafisk brukergrensesnitt
<i>panels</i>	Paneler deler inn GUI i flere overordnede deler.
<i>modelling panel</i>	Hovedpanelet i GUI. Brukes til å lage og vise konstruksjons- og beregningsmodell og til å vise resultater av analyser.
<i>left panel</i>	Sidepanel, inneholder tabeller med <i>joint data member data</i> . <i>Joint</i> -koordinater kan endres i <i>joint data</i> .
<i>bottom panel</i>	Bunnpanel, inneholder feilmeldinger og informasjon fra analyser.
<i>toolbox panel</i>	Verktøykasse, et panel som brukes til å manipulere modellvinduet.
<i>application menu</i>	Applikasjonsmeny, øverst i venstre hjørne av GUI. Brukes til filbehandling og innstillinger.
<i>quick access toolbar</i>	Hurtigverktøylinje, med <i>buttons</i> for innstillinger, <i>undo/redo</i> , lagring og utskrift.
<i>status bar</i>	Statuslinjen, nederst i GUI. Viser koordinatene til musepekeren og en <i>button</i> for utskrift.



<i>ribbon</i>	Horisontalt bånd øverst i grensesnittet som brukes til å organisere <i>ribbon pages</i> .
<i>ribbon page</i>	Fane som brukes til å organisere <i>ribbon page groups</i> . En fane representerer en hovedfunksjon i programmet, i vårt tilfelle <i>modelling, loading, analysis</i> og <i>results</i> .
<i>ribbon page group</i>	Gruppe innad i en fane som brukes for å samle <i>controls</i> som hører logisk sammen.
<i>controls</i>	Objekter som brukes i GUI, eksempelvis i <i>ribbon page groups</i> og dialogbokser. Knapper/ <i>buttons</i> og tekstbokser/ <i>text edits</i> er gode eksempler.
<i>events</i>	Når brukeren benytter en <i>control</i> vil ulike <i>events</i> aktiveres avhengig av hva brukeren gjør med en <i>control</i> . Å trykke på en knapp er en typisk <i>event</i> . Programmet utfører en handling basert på tastetrykket, for eksempel kan det føre til at en dialogboks åpnes.
<i>grid</i>	Rutenett som brukes til å tegne opp <i>members</i> og <i>joints</i> på en presis måte.
<i>grid spacing</i>	Den vertikale og horisontale avstanden mellom linjene i rutenettet.
<i>snap to grid</i>	Verktøy som gjør at <i>joints</i> kan festes til hjørner i rutenettet. Gjør det enklere å modellere presist.
<i>undo</i>	Angrer den siste endringen som er gjort i modellen.
<i>redo</i>	Opphever en <i>undo</i> -operasjon.
<i>popup menu</i>	Menyen som vises når man høyreklikker på <i>structural items</i> . Også kalt <i>right-click menu</i> .
<i>screen coordinate system</i>	Det konstante koordinatsystemet som beskriver pikslenes plassering på en PC-skjerm.
<i>world coordinate system</i>	Det varierende koordinatsystemet som brukes for å beskrive størrelsen og plasseringen av konstruksjoner i modellverdenen.
<i>modelling display</i>	Modellvisningen, modellvinduet.
<i>results display</i>	Resultatvisningen, resultatvinduet.

Tabell 2.4: Engelske uttrykk som beskriver konstruksjonsmodell.

**Konstruksjonsmodell**

Engelsk	Norsk
<i>structural items</i>	Konstruksjonsdel. Eksempelvis bjelker og søyler, knutepunkter, ledd og randbetingelser.
<i>joint</i>	Knutepunkt. Knytter sammen <i>members</i> .
<i>member</i>	Bjelke, søyle, stav, kabel o.l.
<i>internal joint</i>	<i>Joint</i> som er plassert inne på en <i>member</i> . <i>Internal joints</i> deler <i>members</i> inn i <i>submembers</i> .
<i>host member</i>	<i>Member</i> som er vert for interne <i>joints</i> og <i>submembers</i> .
<i>submember</i>	<i>Submembers</i> utgjør tilsammen én <i>member</i> . <i>Submembers</i> kan påføres ulike laster og benytte ulikt antall elementer og i noen grad ulike <i>cross sections</i> .
<i>boundary condition</i>	Grense- (rand)-betingelse. Låser spesifiserte forskyvningsfrihetsgrader i et <i>joint</i> .
<i>support</i>	Opplager, form for randbetingelse.
<i>hinge</i>	Leddforbindelse i knutepunkt. Rotasjon er frikoblet mellom <i>members</i> .
<i>x-dof release</i>	Knutepunktsforbindelse der translasjon i (lokal) x-retning er frikoblet mellom <i>members</i> .
<i>coupling spring</i>	Fjær som innfører motstand mellom frakoblede frihetsgrader i <i>joints</i> der det er <i>hinges</i> eller <i>x-dof releases</i> .
<i>eccentricity</i>	Eksentrisitet. En del av en <i>submember</i> som er helt stiv. Kan påføres i endene av <i>submember</i> -en.
<i>point load</i>	Punktlast (og moment). Settes på i <i>joints</i> eller langs <i>members</i> (hvor det så oppstår <i>internal joints</i> ).
<i>line load</i>	Fordelt last. Fordeles langs valgt <i>submember</i> . Kan variere lineært.
<i>cross section</i>	Tverrsnitt. Påføres i endene av hver <i>submember</i> og kan derfor variere lineært langs <i>submember</i> -en (for noen tverrsnittstyper).
<i>load case (LC)</i>	Laster som virker samtidig samles i et lasttilfelle. Eksempelvis egenvekt av konstruksjoner.
<i>load combination (LCmb)</i>	Lastkombinasjon. Lasttilfeller kombineres for å analysere ulike last-scenarier.

Tabell 2.5: Engelske uttrykk som beskriver beregningsmodell og analyser.

<b>Beregningsmodell</b>	
Engelsk	Norsk
<i>computational model</i>	Beregningsmodell basert på konstruksjonsmodellen og elementmetoden for rammekonstruksjoner.
<i>mesh</i>	Inndelingen av ( <i>sub</i> ) <i>members</i> i elementer. Størrelsen til et <i>mesh</i> er det samme som antall elementer.
<i>mass model</i>	Massemodell for konstruksjoner, på matrisiform. Det finnes ulike modeller.
<i>lumped mass</i>	Samler massen til knutepunktene translasjonsfrihetsgrader.
<i>consistent mass</i>	Fordeler massen analogt til stivheten. Alle frihetsgrader får masse, samt masse på koblingsleddene.
<i>diagonalized mass</i>	En kombinasjon av de to andre modellene. Tar hensyn til rotasjoner, men det er ingen kobling mellom translasjons- og rotasjonsmassen.
<i>load lumping</i>	Metode som gjør fordelte laster om til statisk ekvivalente konsentrerte laster.
<i>“real time” analysis</i>	Sanntidsanalyse, i den forstand at man analyserer <i>load trains</i> som flyttes langs en <i>load path</i> i en konstruksjon.
<i>load trains</i>	Lasttog, et sett med punktlaster som beveger seg samlet langs en <i>load path</i> .
<i>load path</i>	Lastbane, en sti langs en konstruksjon som lasttog forflytter seg etter stegvis.
<i>mode shape</i>	Forskyvningsmønster koblet til en egenfrekvens eller en knekkform.

## 2.2 Programmet

**fap2D** består av et grafisk brukergrensesnitt - GUI, og en beregningskjerne - **Frame2D**. Brukerens oppgave er i første omgang å lage en modell av en konstruksjon vha. GUI. Etter at det er gjort kjøres en valgt analyse av konstruksjonen og til slutt vurderes resultatene av denne.

Beregningskjernens oppgave er å utføre numeriske beregninger, etter forskyvningsmetoden på matriseform, på den beregningsmodellen som GUI serverer.

### 2.2.1 Brukergrensesnitt - GUI

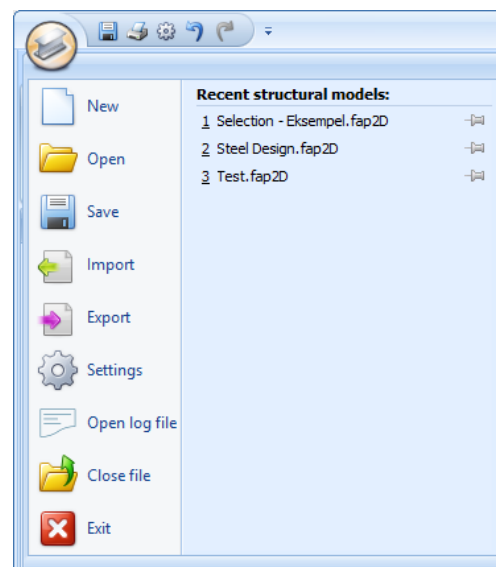
Filosofien for GUI er at det skal være intuitivt og enkelt. Det første som møter brukeren ved oppstart er velkomstscreenskjermen som gir valget mellom å lage en ny modell eller åpne en eksisterende modell.



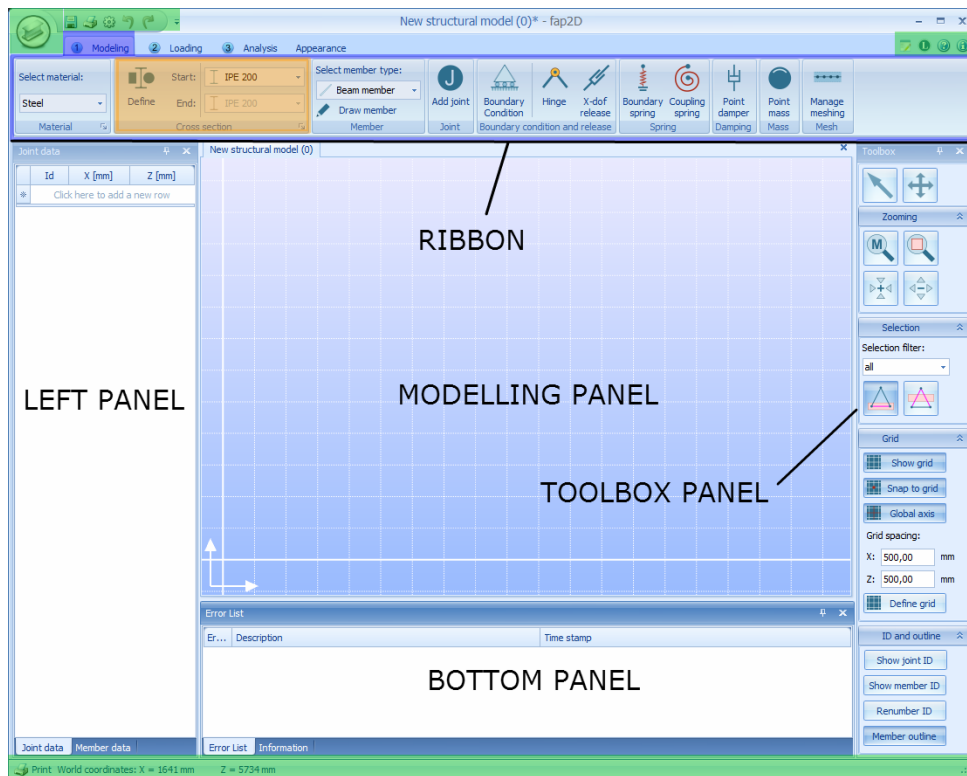
Figur 2.1: Velkomstscreenskjermen.

Velkomstscreenskjermen fører en videre til selve brukergrensesnittet. Det er bygd opp av en *ribbon* og fire paneller. *Ribbon* er det horisontale båndet øverst i GUI. Det inneholder *application menu*, *quick access toolbar* og *ribbon pages*.

*Application menu* brukes til filbehandling og *settings*. *Quick access toolbar* gir raskere tilgang til funksjoner for filbehandling, *settings* og *undo/redo*. Øverst til høyre i *ribbon* finnes knapper som gir tilgang til blant annet lisensavtale og brukermanual.



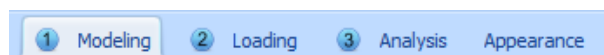
Figur 2.2: *Application menu* og *quick access toolbar* (øverst).



Figur 2.3: GUI i **fap2D**. Merket i grønt: *Application menu* og *quick access toolbar* (øverst t.v), UM, lisensavtale o.l. (øverst t.h) og *status bar* (nederst t.v).

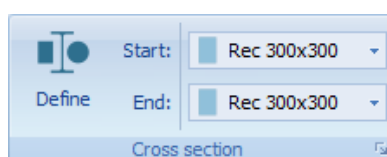
*Ribbon* brukes for å organisere *ribbon pages*. *Ribbon pages* er den groveste inndelingen av funksjonalitet. Innenfor hver *page* samles funksjonalitet som i fellesskap utfører en overordnet oppgave. Eksempelvis modellering av konstruksjoner. Innenfor én enkelt *page* deles funksjonaliteten opp i grupper, såkalte *ribbon page groups*. Det oransje rektangelet i figur 2.3 representerer en gruppe. Én enkelt funksjon er representert ved en *control*. Knapper er typiske *controls*.

I **fap2D** finnes følgende *ribbon pages*: 1 *Modelling*, 2 *Loading*, 3 *Analysis*, 4 *Results* og *Appearance*. 4 *Results page* er et eksempel på en kontekstbasert *ribbon page*. Kontekstbaserte objekter vises bare når en bestemt *event* forekommer, i dette tilfellet når en analyse er ferdig slik at resultater kan vises. Det finnes også kontekstbaserte grupper og knapper i **fap2D**.



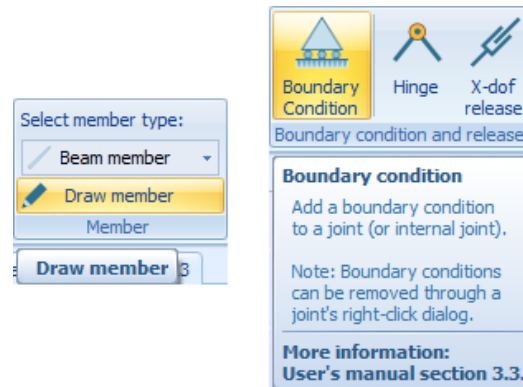
Figur 2.4: *Ribbon pages*. Merk at 4 *Results* er skjult (kontekstbasert).

*Cross section* er et eksempel på en *ribbon page group*. Den inneholder en vanlig knapp og to nedtrekksknapper som lar oss definere og velge tverrsnittene som skal brukes i modellen.



Figur 2.5: Gruppen *Cross section* med tilhørende *controls*.

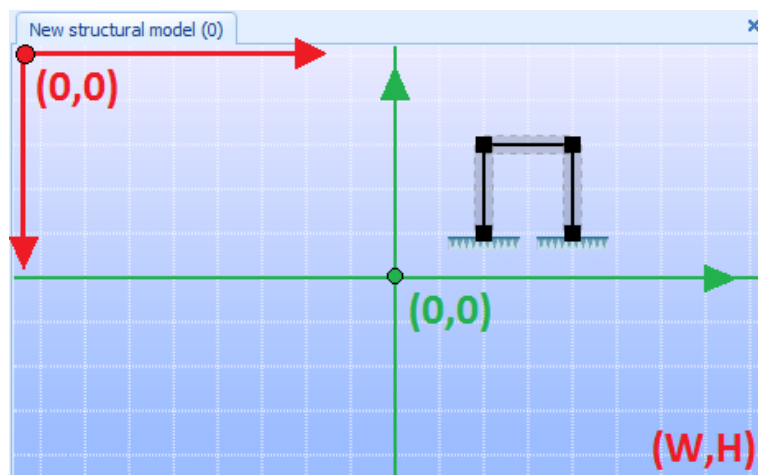
For å gi brukeren mer informasjon om de ulike funksjonene aktiveres *tooltips* og *super tooltips* når man holder musepekeren over en knapp (*control*).



Figur 2.6: F.v.: *Tooltip* og *super tooltip*

Panelene i grensesnittet består av *modelling panel*, *toolbox panel*, *left panel* og *bottom panel*. Alle bortsett fra førstnevnte kan skjules og flyttes på.

Modellvinduet (*modelling panel*) inneholder “tegnearket” som konstruksjonsmodellen lages i. Panelet inneholder to ulike visninger (*displays*), nemlig modellvisningen (*modelling display*) og resultatvisningen (*results display*). Det ene brukes altså til å modellere, mens det andre brukes for å vise resultater. Førstnevnte er aktivt for *modelling*, *loading* og *analysis ribbon pages*, dvs. for alt som skjer før en analyse blir kjørt. Sistnevnte er aktivt etter at en analyse er gjennomført, altså i *results ribbon page*.



Figur 2.7: *Screen coordinate system* (i rødt) og *world coordinate system* (i grønt) i modellvinduet.

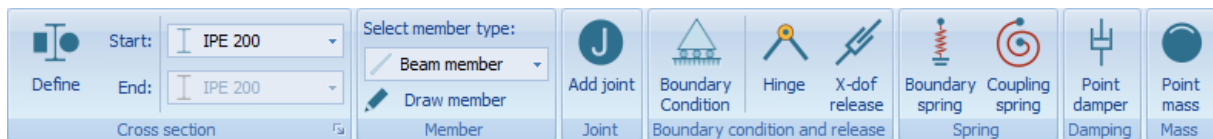
I det statiske *screen coordinate system* defineres objekter på skjermen vha. piksler, dvs. at objektene har konstant størrelse. I det foranderlige *world coordinate system* defineres objekter vha. de fysiske dimensjonene i “modellverdenen”, dvs. at størrelsen deres kan endre seg (ved å zoome).

*Left panel* inneholder informasjon om *members* og *joints*, mens *bottom panel* viser feilmeldinger, advarsler og informasjon fra analyser. *Toolbox panel* inneholder verktøy for å manipulere *structural items* i modellvinduet. Blant annet finnes verktøyene *zoom*, *selection filter* og *grid*.

For mer informasjon om brukergrensesnittet henvises det til tabell 2.3 og til UM. [7]

### 2.2.2 Konstruksjonsmodell og laster

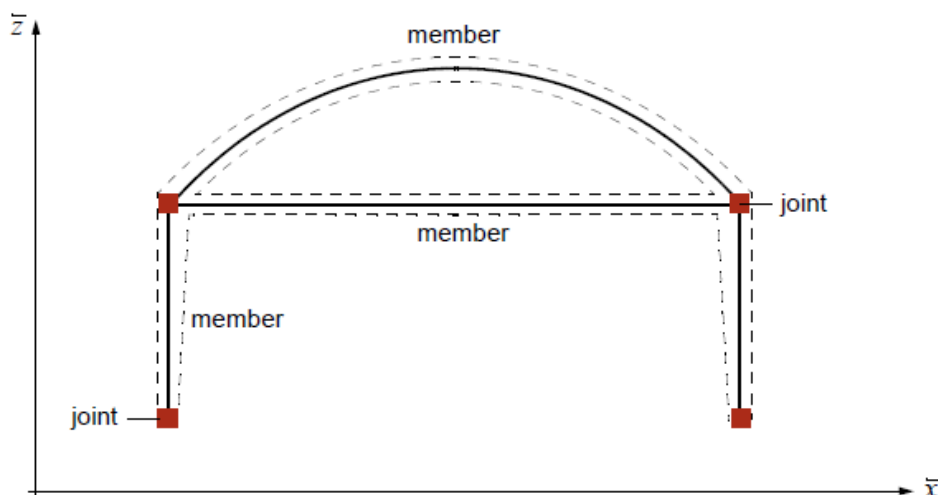
En konstruksjonsmodell i **fap2D** består av *structural items*. I hovedsak vil det si *members* og *joints*. *Members* kan deles i et antall *submembers* eller nye *members*. *Submembers* knyttes sammen av *internal joints*, mens *members* knyttes sammen av *joints*. I *joints* og *internal joints* kan det settes på randbetingelser, fjærer, punktdempere og punktmasser. Figur 2.8 viser alle *structural items* i programmet, med unntak av laster.



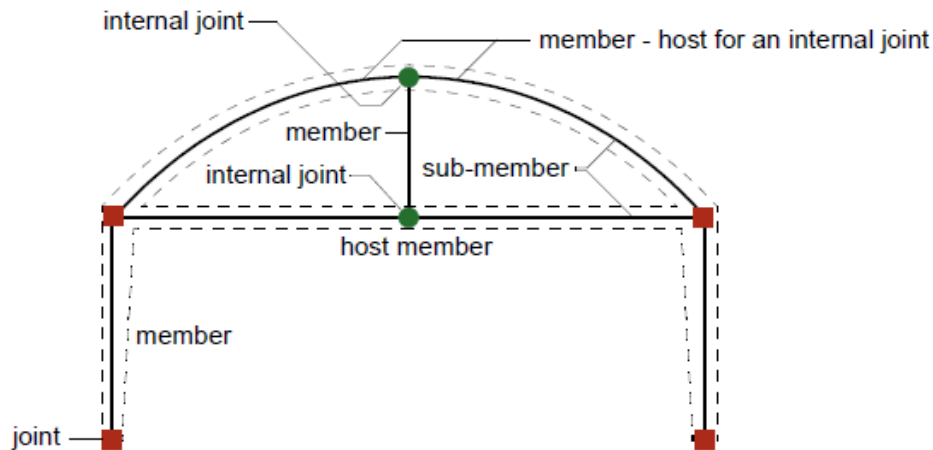
Figur 2.8: *Structural items*.

**fap2D** er laget for å modellere en-dimensjonale konstruksjoner slik som søyler og bjelker. Det er mulig å lage stive partier vha. eksentrisiteter. Eksentrisiteter er masseløse.

Å modellere helt stive deler vha. *members* med tilnærmet uendelig stivhet anbefales ikke, det kan føre til en dårlig kondisjonert stivhetsmatrise som kan gi numeriske problemer. Se avsnitt 7.3 i [4] for mer informasjon.



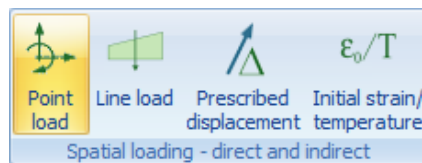
Figur 2.9: Hentet fra figur 1 i UM. [7]



Figur 2.10: Hentet fra figur 2 i UM. [7]

De ulike lasttypene i **fab2D** vises i figur 2.11. Punktlaster/momenter kan settes ut i *joints* eller *internal joints*. De kan også settes ut inne på *members*, men det fører automatisk til at et *internal joint* opprettes i det belastede punktet.

Linjelaster er fordelte laster som settes på *members* eller *submembers*. Linjelaster kan settes på som gravitasjons-, projeksjons-, normal- og aksiallast. Frihetsgradene i *joints* kan påføres foreskrevne forskyvninger som resulterer i indre krefter i statisk ubestemte konstruksjoner. Den siste lasttypen er initialtøyninger (for eksempel pga. temperatur), som kan påføres på *members* og *submembers*. Dersom en konstruksjon er statisk ubestemt vil det oppstå tvangskrefter på grunn av initialbetingelsene.



Figur 2.11: Lasttyper. [7]

For mer utdypende informasjon om konstruksjonsmodellen se s. 8-18 i UM. [7]

### 2.2.3 Beregningsmodell

Konstruksjonsmodellen, som brukeren forholder seg til under modelleringen, blir konvertert til en beregningsmodell når en analyse kjøres.

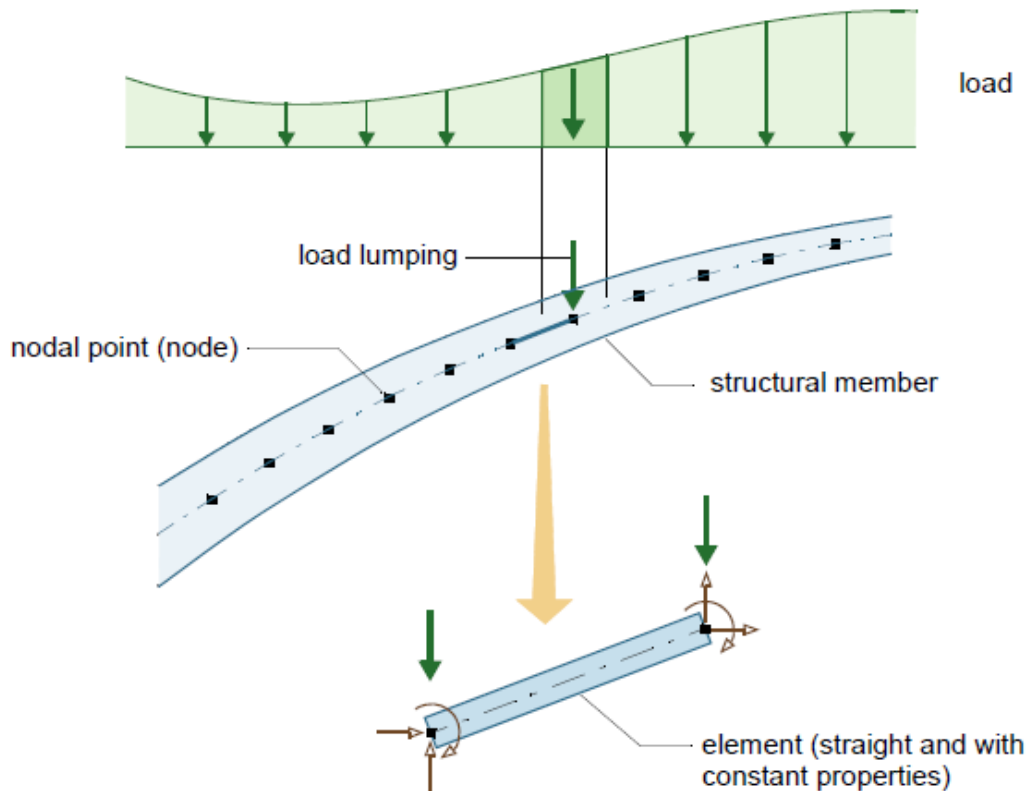
*Members* deles inn i et antall rette elementer. Brukeren kan spesifisere antallet elementer, eller bruke standardverdiene som er satt i programmet. Siden *arch members* også deles inn i rette elementer er det viktig at elementene ikke er for lange så buen i *member*-en blir presist modellert. Dessuten er det oppdaget en feil i elementdefinisjonen for *arch members* som gjør at det er enda viktigere å bruke korte elementer. Vedlegg A.5.4 forklarer dette.

Hvert element har tre frihetsgrader per node og konstante egenskaper. De konstante egenskapene beregnes midt i elementet, eksempelvis tverrsnittsareal og 2. arealmoment.



Frihetsgradene for hver node består av translasjoner i global x- og z-akse, samt rotasjon om y-aksen (ut av planet).

Dersom et element er påført en fordelt last vil lasten omformes til statisk ekvivalente, konsentrerte nodelaster vha. *load lumping*.



Figur 2.12: Beregningsmodellen i **fap2D**, hentet fra figur 12 i UM [7]:

Det finnes tre typer beregningselementer i **fap2D**:

1. *Beam element* - bjelkeelement med seks frihetsgrader, som tar hensyn til aksial- og bøyedeformasjoner, og skjærdeformasjoner hvis det er ønskelig.
2. *Axial element* - stav-, kabel- og *strut*-elementer der rotasjonsfrihetsgradene er deaktivert slik at det bare tas hensyn til aksialdeformasjoner. Kabler tar strekk, *struts* tar trykk og staver tar både strekk og trykk.
3. *Spring element* - Fjærelementer, både rotasjons- og translasjonsfjærer. Kan settes på som en randbetingelse for én node eller som en koblingsfjær mellom to ulike noder/elementer. Både lineære og ikkelineære fjærer er tilgjengelige.

Etter at alle elementer er etablert blir de satt sammen på en måte som resulterer i et system av lineære ligninger. Foreskrevne frihetsgrader fjernes fra ligningssystemet før det løses. Analysetypen bestemmer hvordan modellen skal løses, men felles for de fleste analysene er at de benytter seg av en direkte ligningsløser basert på  $LDL^T$  faktorisering.<sup>1</sup> For

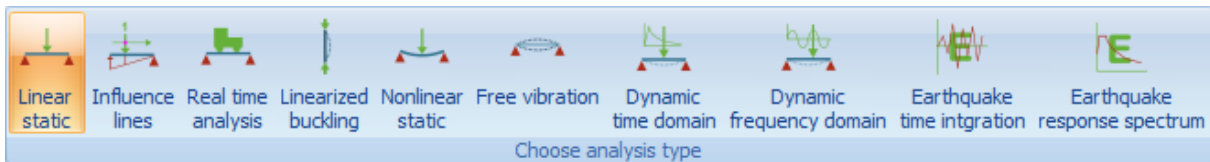
<sup>1</sup>Metoden er beskrevet på s. 243-247 i [4].

egenverdi problemer som fri svingning og knekning brukes en iterativ egenverdi løser, som benytter direkte ligningsløsning.

For mer informasjon om beregningsmodellen henvises det til UM [7, s. 19-22].

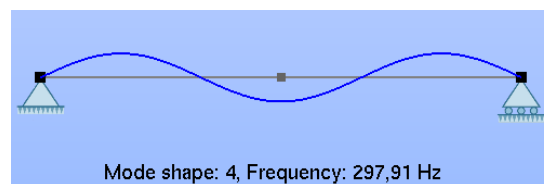
## 2.2.4 Analysetyper og resultater

Analysene i **fap2D** kan grovt sett deles inn i statiske og dynamiske analyser. Figur 2.13 viser knapper for alle analysetypene. I tillegg er det mulig å kontrollere konstruksjoner i hht. til ståldimensjonering i *Eurocode 3-1-1* [2].



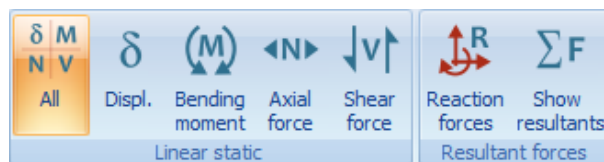
Figur 2.13: Analysetypene i **fap2D**.

Resultater fra analysene presenteres i en rekke ulike visninger og diagrammer, slik at brukeren kan vurdere konstruksjoner for dimensjonering i både bruks- og bruddgrensetilstand. Resultatvisningene er avhengige av analysetype. Figur 2.14 viser f.eks. resultater for fri svingning i form av egenfrekvenser og tilhørende *mode shapes*.



Figur 2.14: Resultatvisning for fri svingning.

For de fleste andre analysene vises responsdiagrammene i hht. knappene i figur 2.15. Dvs. forskyvnings-, moment-, aksial- og skjærkraftdiagram. I tillegg vises resultant- og reaksjonskrefter for de statiske analysene og for *real time analysis*.



Figur 2.15: Knapper knyttet til resultatdiagrammer.

For mer utdypende informasjon om analyser henvises det til UM [7, s. 28-65]. De analysene som er mest relevante i denne rapporten blir introdusert nærmere i senere kapitler.

# Kapittel 3

## Endringer i fap2D

Version 3.0.0 - Beta → Version 3.1

Dette kapitlet gir innblikk i alle endringene som er innført i den nye versjonen av **fap2D**, versjon **3.1**. Hoveddelen av masteroppgaven har gått med til å utføre endringene som er beskrevet her.

Endringene har blitt klassifiserte som enten feilrettinger, modifikasjoner eller ny implementasjoner. Feilretting fjerner opplagt svakheter i programmet, som ikke burde vært tilstede. Modifikasjoner er en mindre justeringer av eksisterende kode, som er ment å forbedre programmet. En ny implementasjon er en fullstendig ny funksjonalitet. Den sistnevnte typen endringer er gjerne mest synlig for brukeren, mens førstnevnte som regel er mest arbeidskrevende fordi feilsøking er krevende.

Detaljerte beskrivelser av innholdet i dette kapitlet finnes i vedlegg A *Release notes* - Versjonsmerknader. Det er ikke ment at vedlegges skal leses fram perm til perm, det bør heller brukes som et oppslagsverk når det er nødvendig med mer informasjon. Strukturen i vedlegget er i hovedsak den samme som her.

### 3.1 GUI

Brukergrensesnittet i **fap2D** består i hovedsak av to komponenter. Det første er rammeverket DevExpress (DX) som er brukt for å lage det overordnede designet og alle *controls* i programmet. Alle panelene og de aller fleste dialogboksene er f.eks. basert på DX. Den andre komponenten i GUI er grafikk-rammeverket OpenGL som presenteres i avsnitt 3.2 Tegning - OpenGL.

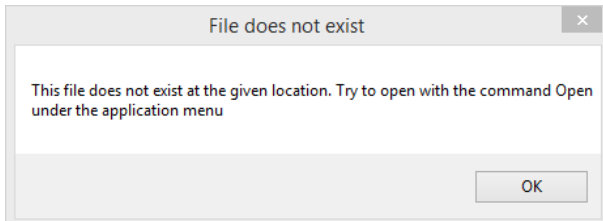
Detaljer om endringene i dette avsnittet er gitt i vedlegg A.1.

#### 3.1.1 Design

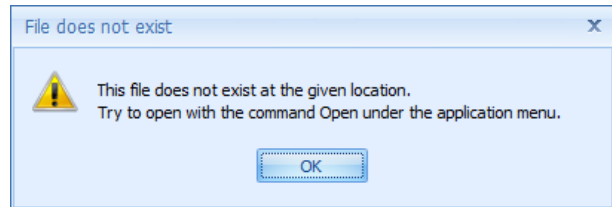
**fap2D** skal være brukervennlig og derfor er det en fordel om programmet har en profesjonell og oversiktlig layout. De kosmetiske inngrepene som er gjennomført for å oppnå dette beskrives her.

## Feilretting av designobjekter

Noen dialogbokser og tekstfelter som var Windows Forms (WF)-objekter er gjort om til DX-objekter, slik at så mange objekter som mulig har samme design. GUI framstår dermed mer helhetlig.

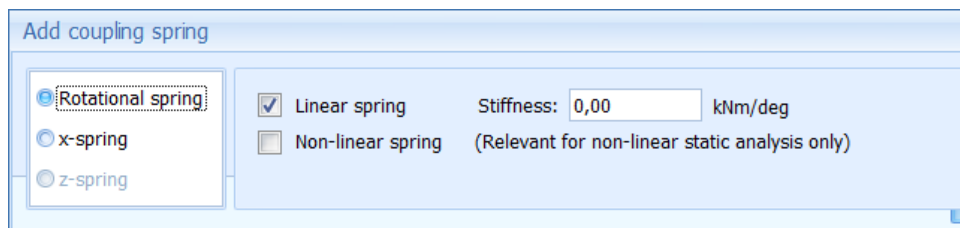


Figur 3.1: Versjon 3.0: WF-dialogboks.

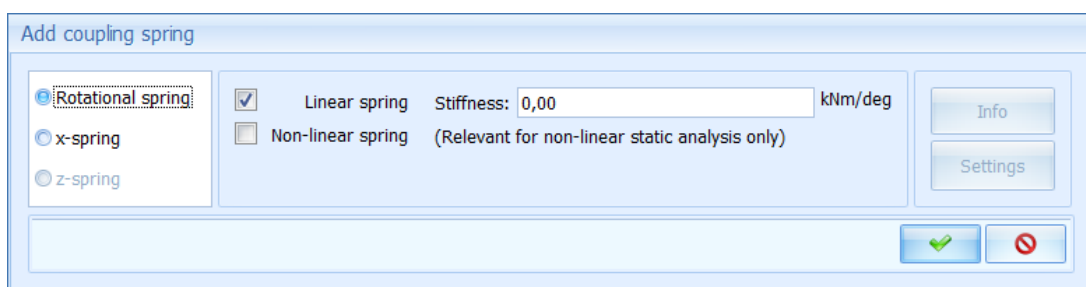


Figur 3.2: Versjon 3.1: DX-dialogboks.

Dialogbokser med flere *controls* ble designet med utgangspunkt i at tekst og symboler hadde fast størrelse, selv om det viser seg at de kunne ekspandere. Ekspansjonen gjorde at enkelte dialogbokser fikk unaturlig utseende, som også kunne påvirke funksjonaliteten deres. Dette er rettet ved å benytte `XtraLayoutControl` fra DX som er utviklet for å tilpasse layout av dialogbokser bedre.



Figur 3.3: Versjon 3.0: Noen knapper er skjult i dialogboksen for fjærer (i *modelling ribbon page*).



Figur 3.4: Versjon 3.1: Alle knappene i dialogboksen for fjærer vises.

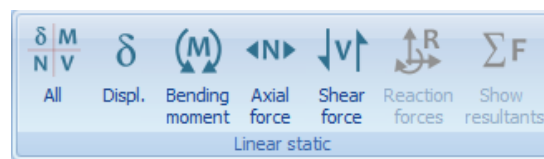
Samme dialogboks brukes ved opprettelse og redigering av linjelaster. Knappen for å godkjenne valgene i denne var opprinnelig et plusstegn i begge tilfellene, men det var misvisende fordi plusstegn symboliserer at noe nytt legges til. Ved redigering vises det derfor et hakesymbol i versjon 3.1, siden man kun bekrefter endringer for en last.



Figur 3.5: Knapper med plusstegn og hakesymbol for hhv. opprettelse og redigering av *structural items*.

I grafen for ikke-lineær fjærstivhet, i dialogboksen for fjærer, ble det funnet en feil som nå er rettet. Feilen var at y-aksen på grafen ikke ble oppdatert når man vekslet mellom aksial- og rotasjonstivhet. Slike små rettinger virker ubetydelige, men fjerner rom for misforståelser.

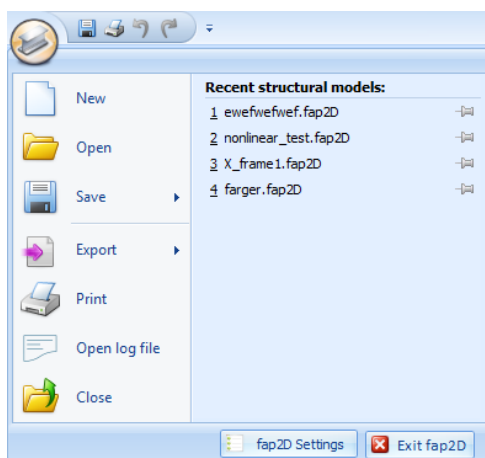
Diagrammet som viser reaksjonskrefter og dialogen som viser resultatantkraftene i en konstruksjon er ikke aktive for dynamiske analyser. Derfor har knappene i figur 3.6, som brukes til å vise diagrammet og dialogen, blitt skjult fra *ribbon* siden de er unødvendige.



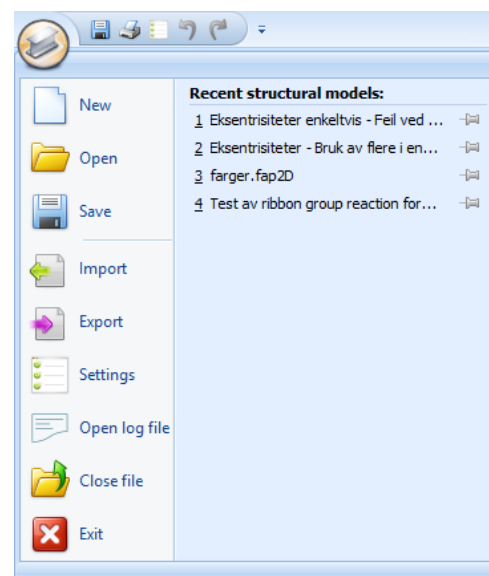
Figur 3.6: Versjon 3.0: Knapper for resultat- og reaksjonskrefter er deaktivert for dynamiske analyser.

### Modifiserte og nye designobjekter

Layouten i *application menu* er endret slik at *settings-* og *exit-*knappene nå er plassert sammen med de andre knappene. Tidligere var det vanskeligere å legge merke til knappene, siden de var små og ukonvensjonelt plassert. Knappen *Print* er fjernet og erstattet av knapper som gir tilgang til de nye eksport- og import-funksjonalitetene som presenteres i avsnitt 3.8.



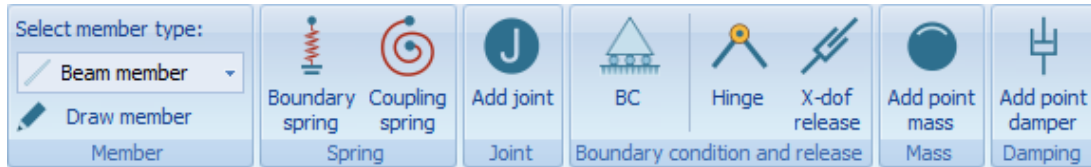
Figur 3.7: Versjon 3.0: *Application menu*.



Figur 3.8: Versjon 3.1: *Application menu*.

I noen tilfeller har *controls* blitt flyttet til posisjoner som gir mer mening. Siden modelleringsprosessen i **fap2D** foregår fra venstre mot høyre er rekkefølgen av knappene for *struc-*

*tural items* i *modelling ribbon page* justert slik at *items* som brukes oftere er flyttet lengre mot venstre. Dvs. at *joints* og grensebetingelser er flyttet i forhold til posisjonen de hadde i figur 3.9. Dette passer også bedre fordi fjærer, dempere og masser alle brukes i dynamiske analyser og bør stå ved siden av hverandre, slik de nå gjør i figur 3.10.



Figur 3.9: Versjon **3.0**: Plassering av knapper for *items* (i *modelling ribbon page*).



Figur 3.10: Versjon **3.1**: Plassering av knapper for *items* (i *modelling ribbon page*).

For å gi mer konsistent layout har *zooming*-gruppen blitt flyttet over *selection*-gruppen i *toolbox panel*. Grunnen er at *zooming* er synlig i både modellerings- og resultatvisningen, og gruppen ble tidligere forskjøvet opp og ned i panelet når man byttet visning.

Som figur 3.11 viser er symbolet for den nevnte *settings*-knappen blitt skiftet ut til fordel for et mer familiært og beskrivende symbol, i form av et tannhjul. Symbolet er bevisst valgt fordi det ser tydelig ut selv når det skaleres ned i *quick access toolbar*.

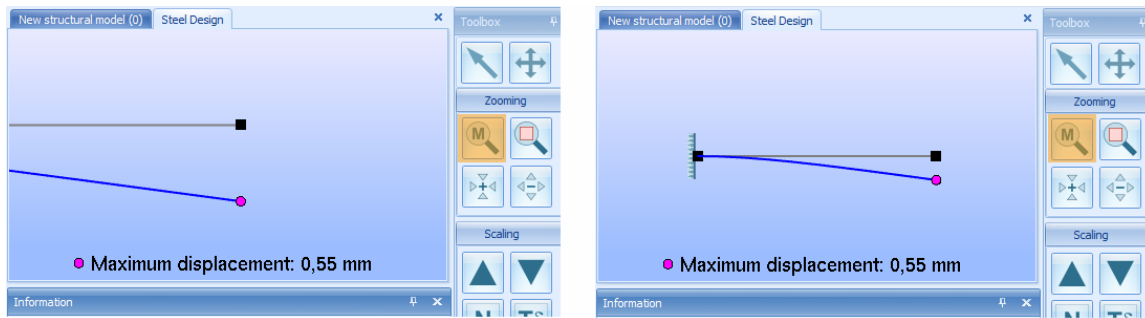


Figur 3.11: Gammelt (t.v) og nytt symbol (t.h) for *settings button*.

### 3.1.2 Oppførselen til *controls*

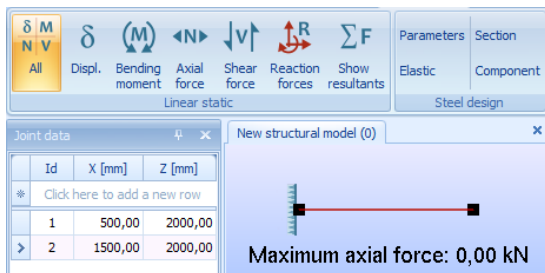
Dette avsnittet tar for seg modifisering av *controls* som ikke oppførte seg intuitivt og korrigerende av *controls* som oppførte seg direkte feil.

Etter at en analyse er gjennomført skal resultatene av den vises. En praktisk endring som er innført i versjon **3.1** er at programmet automatisk veksler til resultatvisningen, i motsetning til i versjon **3.0** der man måtte veksle manuelt. I den nye versjonen vil dessuten zoom-nivået i modellen bli justert i det en analyse er ferdig, slik at den presenteres pent i resultatvinduet. Det er lagt til en innstilling *settings* der man kan slå av denne zoom-funksjonen og heller beholde samme zoom-nivå som før analysen.

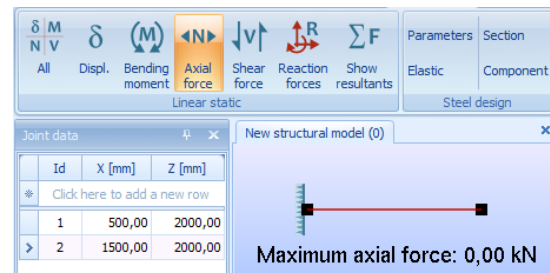


Figur 3.12: Versjon **3.0** (t.v): Hele modellen vises ikke i resultatvinduet -  
Versjon **3.1** (t.h): Hele modellen vises pga. *zoom to model*

Knappene som veksler mellom de ulike visningene av resultatdiagrammer samsvarer nå med diagrammet som vises, noe de ikke gjorde før. En tydeligere forklaring er gitt i vedlegg A.1.12 og i figur 3.13 og 3.14 nedenfor:

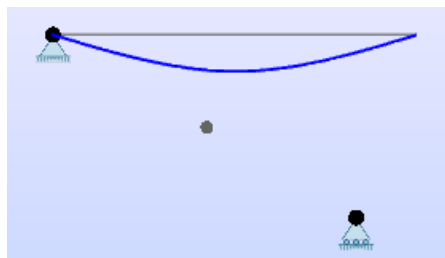


Figur 3.13: Versjon **3.0**: Ikke samsvar mellom diagram og knapp.



Figur 3.14: Versjon **3.1**: Samsvar mellom diagram og knapp.

En feil som gjorde at programmet kun regnet på én av flere åpne modeller er rettet opp. Under analyse skiftet programmet fokus til en annen modell dersom det var utført analyser på flere av dem. I tillegg er det ikke lenger mulig å vise utdaterte resultater for modeller som har blitt endret i etterkant av en analyse. Figur 3.15 oppsummerer problemstillingen, men feilen er ganske spesifikk så det anbefales å lese vedlegg A.1.14 dersom man er interessert i detaljene.



Figur 3.15: Versjon **3.0**: Utdatert resultatvisning for en modell som har blitt endret.

Dialogboksen for fjærer kan utvides når man trykker på noen av knappene i dialogen. Tidligere gjorde dette at dialogen ble repositionert til midten av skjermen veldig ofte og det kunne være plagsomt. Det har derfor blitt innført en metode som sørger for at dialogen bare

blir reposisjonert når deler av den havner utenfor skjermen. Figurene i vedlegg A.1.15 viser et eksempel.

En annen dialogboks ble frigjort fra minnet da den ble lukket, noe som gjorde at hele modellpanelet ble nullstilt for så å bli tegnet på nytt igjen. Denne bruker nå `Close()`-metoden, i likhet med andre dialogbokser, slik at lukkingen fremstår som mindre dramatisk.

Tekstfeltene som viser output i form av resultantkrefter kan ikke lenger bli endret av brukeren. Det skal nemlig ikke gå an å endre en output-verdi. Derimot er det fortsatt mulig å kopiere output-verdiene.

Knappene for *member*, *joint* og *boundary conditions* er ikke lenger i konflikt med hverandre. Mer detaljert informasjon finnes i vedlegg A.1.17.

I versjon **3.0** måtte man deaktivere knappene for *member* og *point load* før man kunne aktivere dem igjen og sette ut ny *member/load*. I versjon **3.1** slipper man å deaktivere før de kan brukes på nytt, slik at prosessen blir enklere.

Ellers er en *unhandled exception* for *remove*-knappen i *earthquake loading dialog* nå blitt håndtert slik at programstopp avverges.

### 3.1.3 DevExpress-oppdateringer



For å gjøre stabilisere GUI mest mulig har det blitt lastet ned nye versjoner av DX [17] så fort de har blitt lansert. Deretter har man brukt et konverteringsverktøy for å endre referansene i kildekoden fra det gamle til det nye rammeverket.

Med unntak av én konvertering var overgangen til nye versjoner knirkefri. Det henvises til A.1.21 for detaljer.

## 3.2 Tegning - OpenGL

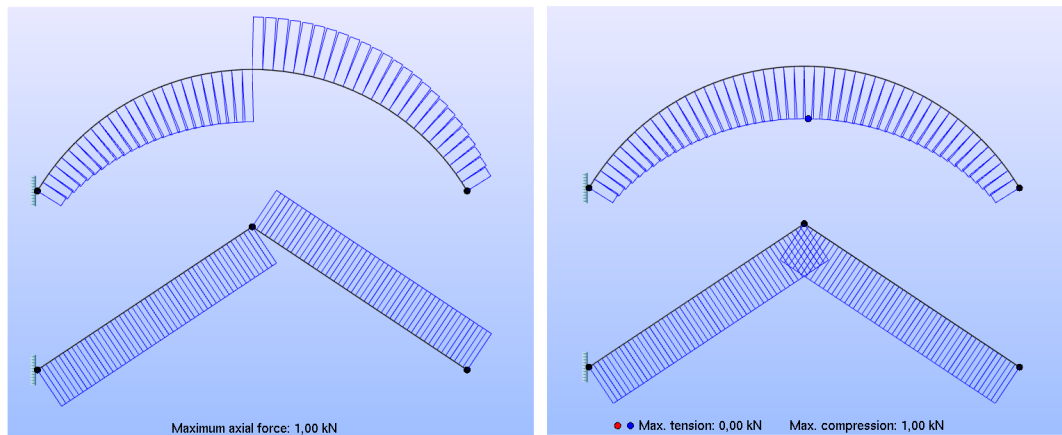
Den andre komponenten i GUI er grafikk-rammeverket OpenGL som i hovedsak brukes til tegning av konstruksjonsmodell og brukerinteraksjon i modellvinduet. Endringer knyttet til OpenGL og tegning presenteres ytterligere i vedlegg A.2.

### 3.2.1 N- og V-diagrammer

Aksial- og skjærkraftdiagrammene (hhv. N- og V-diagrammer) i resultatvisningen ble ved noen tilfeller tegnet på feil side av elementene, og kunne bytte side langs *members*.

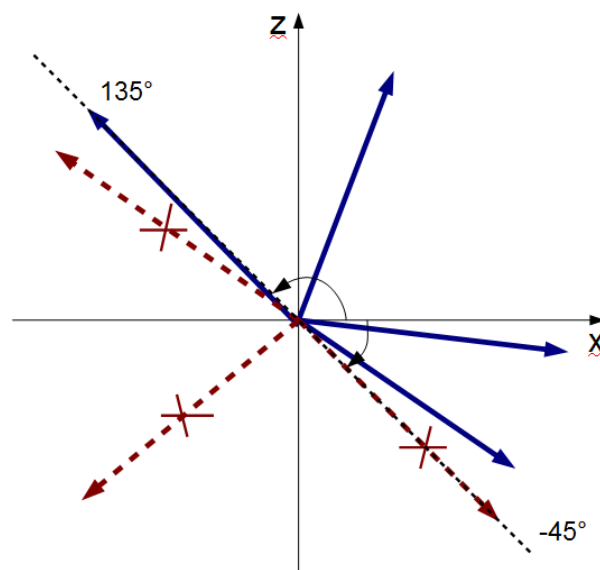
Elementer langs *arch members* tegner nå N- og V-diagrammene på samme måte som bøyemoment slik at positive og negative verdier alltid tegnes på hhv. oversiden og undersiden av elementet. Da unngår man at diagrammet skifter side langs en *arch member*.





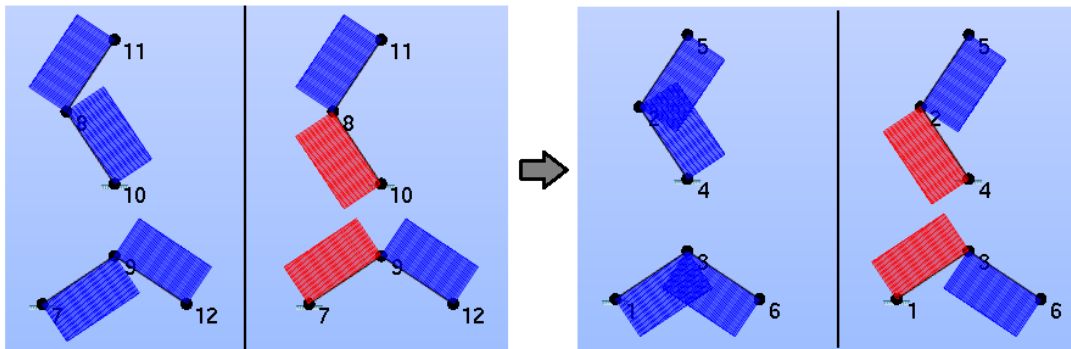
Figur 3.16: Versjon **3.0** (t.v): Aksialkraftdiagrammer som skifter side for en *arch member* og to sammenhengende *members* - Versjon **3.1** (t.h): Aksialkraftdiagrammer tegnes konsekvent for en *arch member* og to sammenhengende *members*.

“Grensevinklene”<sup>1</sup> er forandret fra pluss/minus 180° til å ligge mellom -45° og 135°, som vist i figur 3.17. Denne endringen gjør at diagrammer tegnes på riktig side, både for *members* som er orientert vertikalt og horisontalt, i motsetning til tidligere. Figur 3.18 viser hvordan diagrammene kan oppleves som mer intuitive.



Figur 3.17: Elementer med vinkel fra 135° til -45° blir snudd slik at de blir definert med sin motsatte vinkel. Elementer definert med retning utenfor “grensevinklene” tegnes i motsatt retning.

<sup>1</sup>Vinklene som definerer grensen for retningen på elementene.

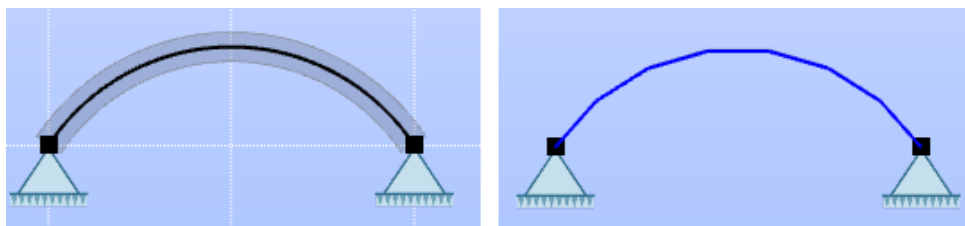


Figur 3.18: Versjon 3.0 (t.v.): N- og V- diagrammer som tilsynelatende skifter side langs en rekke *members* - Versjon 3.1 (t.h.): N- og V- diagrammer som tegnes mer intuitivt.

Dette løser likevel ikke det siste problemet der *members* har nesten lik vinkel som grensevinklene. Derfor er det implementert et søketre som leter seg gjennom *members* som ligger ved grensevinklene og sørger for at de *members* som er koblet sammen med *joints* vil tegnes med kontinuerlige diagrammer. For detaljer se vedlegg A.2.1 Forbedring av N- og V-diagrammer.

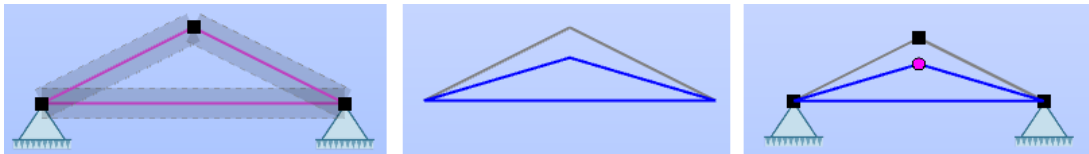
### 3.2.2 Mindre rettinger og modifiseringer av tegnemetoder

Sirkler som bla. brukes til å indikere et valgt element tegnes nå med færre linjestykker for å effektivisere tegnemethodene. Antallet linjestykker har derimot økt fra 30 til 90 for *arch members* slik at tegnes jevnere. Dette er gjort for å skille tydeligere mellom den analytiske definisjonen av *arch members* som brukes i konstruksjonsmodellen, og den diskretiserte elementdefinisjonen som brukes i beregningsmodellen. Forøvrig er metoden for inndeling av tegnepunkter i *arch members* omgjort slik at det deles langs med buen i stedet for korden.

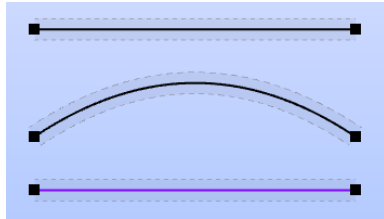


Figur 3.19: Illustrasjon av forskjellen mellom konstruksjonsmodell (glatt, analytisk) og beregningsmodell (diskretisert) for *arch members*.

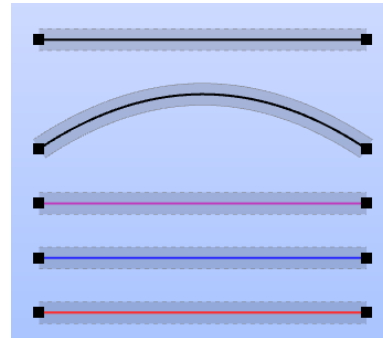
*Joints* og *supports* forbundet med *bar/strut/cable members* blir nå inkludert ved tegning av beregningsmodellen, som vist i figur 3.20. *Bar*, *strut* og *cable members* har i tillegg fått separate farger for å skille dem enklere fra hverandre. I tillegg til svart for *beam* og *arch members*, er *bar members* lilla, *strut members* blå, og *cable members* er røde.



Figur 3.20: Konstruksjonsmodell av et fagverk (t.v) - Versjon **3.0** (i midten): Symboler for *joints* og *supports* blir ikke tegnet i resultatdiagram - Versjon **3.1** (t.h): Symbolene blir nå tegnet.



Figur 3.21: Versjon **3.0**: *Bars*, *cables* og *struts* tegnes med samme farge, og det er svak kontrast mellom *member outline* bakgrunnen.

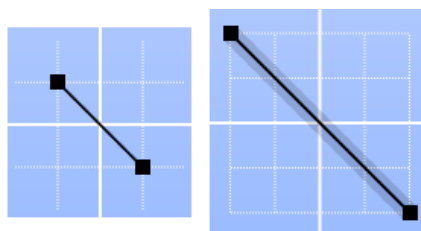


Figur 3.22: Versjon **3.1**: Bruk av flere farger, og *member outline* vises tydeligere.

Fargen på *member outline* har blitt gjort mindre transparent slik at kontrasten til bakgrunnen i modellvisningen blir tydeligere. Dette gjør det også enklere å se overlappende *members*.<sup>2</sup>

Skaleringen av resultatdiagrammene blir ikke lenger tilbakestilt hver gang resultatvisningen skiftes. Den holder seg uendret helt til brukeren benytter seg av skaleringsverktøyet i *toolbox*.

De ytterste linjene i rutenettet (*grid*) ble ikke tegnet i versjon **3.0**, men tegnes nå slik at *joints* kan *snappe* til ytterkantene. *Grid*-funksjonaliteten i **fap2D** gjør at den såkalte *ghost/none member* er overflødig og har blitt fjernet.



Figur 3.23: Versjon **3.0** (t.v): Rutenettet mangler de ytterste linjene - Versjon **3.1** (t.h): Rutenettet tegnes komplett.

<sup>2</sup>Mer om overlappende *members* i avsnitt 3.4.3 og 4.3.1.

### 3.3 Markering av *structural items*

#### Introduksjon

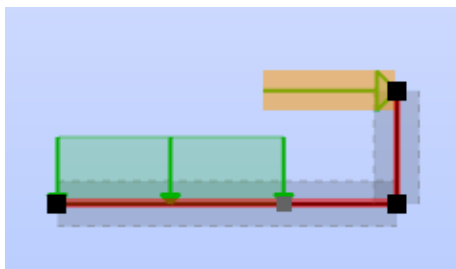
For at det skal være mulig å endre *structural items* etter at de er blitt plassert i modellvinduet, må det være mulig å markere de grafiske objektene som representerer dem. Dette gjøres i **fap2D** vha. metoder som registrerer museklikk og vurderer om de treffer objektene på skjermen. I versjon **3.0** hadde metodene som registrerer markering flere svakheter. Derfor er det utført grundig omlegging av dem. En generalisert forklaring av metodene er gjort i A.3.6. I hovedsak er det de fem *structural items* i tabell 3.1 som kan markeres i modellvinduet.<sup>3</sup>

Tabell 3.1: *Structural items* som kan markeres i *modellering panel*.

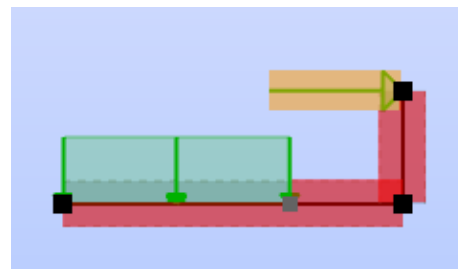
<i>Item</i>	Visning
<i>joints</i> (og <i>internal joints</i> )	modellering, resultat
<i>members</i> (og <i>submembers</i> )	modellering
<i>point loads</i> - punktlaster	modellering
<i>line loads</i> - linjelaster	modellering
Elementer	resultat

#### Definisjon av utstrekningen til grafiske objekter på skjermen

Uttrekningen til de grafiske objektene (som representerer *structural items*) på skjermen er definert slik at det er mulig å markere hvert enkelt av dem når de står nær hverandre. Dessuten skal det være mulig å markere alle *items* så godt det lar seg gjøre når det zoomes langt ut.



Figur 3.24: Versjon **3.0**: Markering av *structural items*.



Figur 3.25: Versjon **3.1**: Markering av *structural items*.

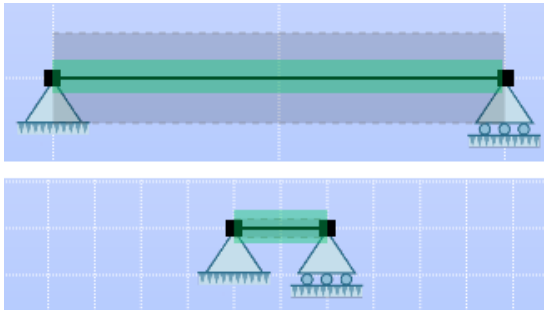
#### 3.3.1 Markering av *members* og *submembers*

##### Rette *members*

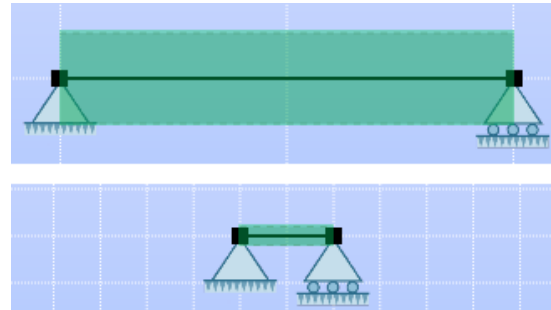
Markeringen av *members* og *submembers* er gjort mer intuitiv og enkel ved å benytte *member outline* som markeringsgrense. Tidligere var grensen gitt i form av pikslene på skjermen,

<sup>3</sup>For å få tilgang til endring av alle de andre *items*, slik som *boundary conditions* og *springs* osv. må man gå via høyreklikk-menyen for *joints* og *members*.

altså måtte man trykke like nær senterlinjen uavhengig av zoom-nivå. Figur 3.26 forsøker å illustrere at dette ikke var intuitivt, men at det alltid var mulig å markere *members* når det ble zoomet langt ut.

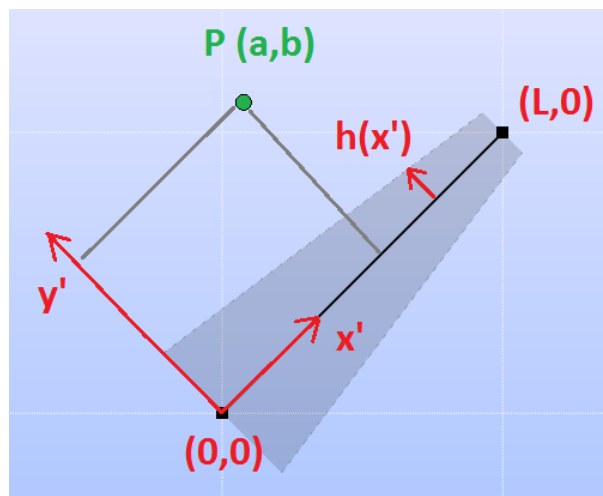


Figur 3.26: Versjon 3.0 (t.v): *Member* markeres når det trykkes tilstrekkelig nær senterlinjen.



Figur 3.27: Versjon 3.1 (t.h): *Member* markeres når det trykkes innenfor *member outline*.

Selv om markeringsområdet kan bli lite i det nye markeringsystemet (se figur 3.27), skal man zoome relativt langt ut for å få problemer med å velge *members*. Konklusjonen er at det nye systemet i praksis er bedre fordi brukeren forventer at det er *member outline* som definerer grensen til objektet.



Figur 3.28: Punktet  $\mathbf{P}$  illustrerer hvor musepekeren har blitt trykket ned. Som man ser ligger punktet utenfor *member outline* og *member*-en i bildet vil ikke bli markert. Ligning 3.1 og 3.2 viser kriteriet som  $\mathbf{P}$  må oppfylle.

For å ta høyde for at *members* kan stilles på skrå og ha varierende tverrsnittshøyde ble det valgt å spesifisere markeringskriteriet for *members* vha. det lokale koordinatsystemet i *member*-en, i stedet for det globale. Det er nemlig mye enklere å spesifisere kriteriet i et lokalt system, slik som vist i figur 3.28.

Kriteriet for å bestemme om et punkt  $\mathbf{P}$  (fra figur 3.28) tilfredsstiller markeringskriteriet er:

$$a < |h(x)| \quad (3.1)$$

$$0 < b < L \quad (3.2)$$

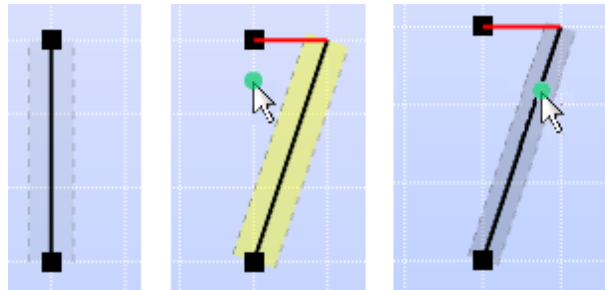
### Arch members

For *arch members* er det vanskeligere å definere utstrekningen enn for rette *members* fordi de defineres av ligninger for sirkler og parabler. Det ble bestemt at *arch members* ikke skal kunne markeres på samme måte som rette *members* fordi det er en beregningsmessig tynge og mer komplisert prosess å gjennomføre. De blir heller markert på den gamle måten der markeringsgrensen er basert på avstand fra senterlinjen.

Likevel er det forsøkt å bøte på problemet ved å legge til en innstilling i *settings dialog* der brukeren kan bestemme hvor nær senterlinjen i en *arch* man må trykke for å markere den. Standardverdien er blitt satt til 15 piksler. Se vedlegg A.9.7 for mer informasjon.

### Members med eksentrisiteter

Eksentrisiteter blir nå tatt hensyn til i markeringsprosedyren for *members*. Figur 3.29 viser et eksempel på at de ikke ble det i versjon 3.0.

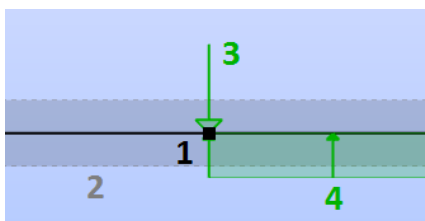


Figur 3.29: Versjon 3.0: Den opprinnelige plasseringen av *member* (t.v) - *Members* med eksentrisiteter måtte markeres ved å trykke langs den opprinnelige plasseringen (i midten) - Forsøket på markering i bildet t.h. fører ikke frem.

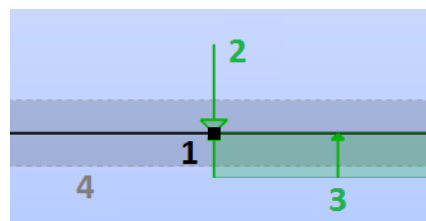
### 3.3.2 Markeringspreferanse

Preferansen for markering av *structural items* er blitt endret; *members* har fått lavere preferanse enn *loads* fordi markeringsgrensen for *members* nå er endret til *member outline* og det gjør at de har et større markeringsfelt enn tidligere på skjermen. Figur 3.24 t.o.m. figur 3.27 viser dette. *Members* har faktisk fått lavest preferanse slik at de mindre objektene kan filtreres ut.

Figur 3.30 og 3.31 poengterer endringen av preferansene.



Figur 3.30: Versjon 3.0: Den gamle markeringspreferansen.



Figur 3.31: Versjon 3.1: Den nye markeringspreferansen.

### 3.3.3 Markering og visning av elementer

Elementer er som regel representert vha. tynne linjestykker på skjermen fordi de gir god plass til tegning av diagrammer. Det eneste unntaket er i *steel design*-diagrammer der de nå tegnes med en tverrsnittshøyde som tilsvarer *member outline*.

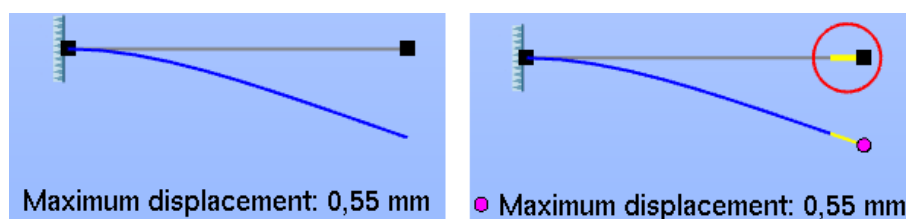
Siden linjestykkene er tynne passer det godt å bruke *screen coordinates* til å definere hvor nært man må trykke for å markere et element. Det er lagt til en innstilling i *settings dialog* der brukeren kan velge akkurat hvor nært.



Figur 3.32: **3.1:** Brukeren kan selv justere hvor presist man må trykke for å markere et element. Figurene illustrerer ulike verdier.

#### Forskjøvet element

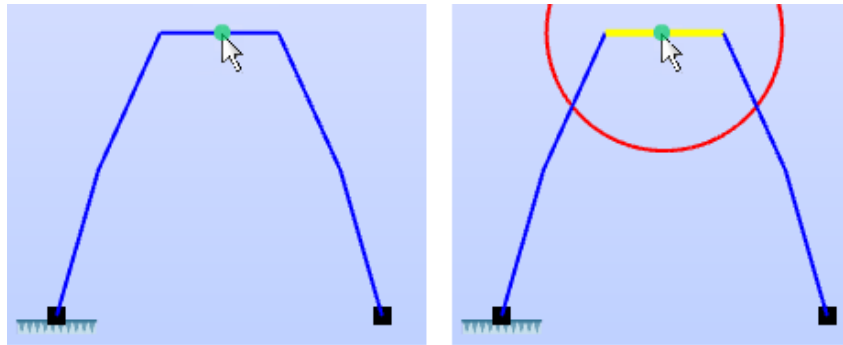
I forskyvningsdiagrammet har flere personer uttrykt ønske om at burde være lettere å finne ut hvor de største forskyvningene opptrer. For å oppnå dette er det nå mulig å markere et element ved å trykke på den forskje versjonen av elementet. Dessuten vil både forskjøvet og original versjon vises som markert i gult når elementet markeres. Sammen med den nye navigeringsfunksjonen i vedlegg A.6.3, og markering av maksimalpunkter i vedlegg A.6.5 har man nå flere gode muligheter for å finne de elementer som er mest forskjøvet. Figur 3.33 viser markeringene:



Figur 3.33: Markering av forskjøvet element (hhv. før og etter), med indikert maksimalverdi for hele utkrageren (i rosa).

#### Markering av elementer i *arch members*

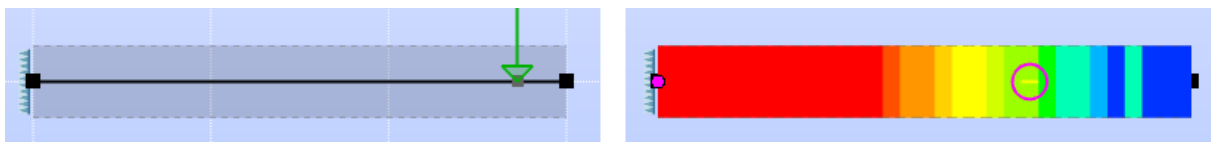
I versjon **3.0** ble det brukt en separat logikk for å markere elementer i *arch members*. Man ser ingen grunn til hvorfor det skulle brukes en separat logikk, siden elementdefinisjonen er den samme for både rette *members* og *arch members*. Derfor ble den enkelt og greit fjernet. Årsaken til at man oppdaget skillet var en feil i prosedyren for *arch member*-elementer som gjorde at det ikke var mulig å markere dem når man klikket langt fra nodene. Dess lengre elementer, desto tydeligere feil.



Figur 3.34: Versjon **3.0** (t.v): Ikke mulig å markere elementer i *arch members* når det trykkes langt fra nodene - Versjon **3.1** (t.h): Feilen er rettet.

### **Steel design usage**

Utnyttelsen av kapasiteten til et **tverrsnitt** kan beregnes vha. standarden for ståldimensjone-ring [2]. Visualiseringen av utnyttelsen har blitt endret slik at resultatdiagrammet tegnes over hele tverrsnittshøyden til elementene. Siden dette er gjort ble det også bestemt at markeringen av elementer i resultatvisningen for *steel design* skal foregå over hele tverrsnittshøyden. Mer om dette i avsnitt 3.7 *Steel design*.



Figur 3.35: Elementene i *steel design*-visningen vises og markeres over hele tverrsnittshøyden til elementene.

### **3.3.4 Feilaktig snap ved museklikk**

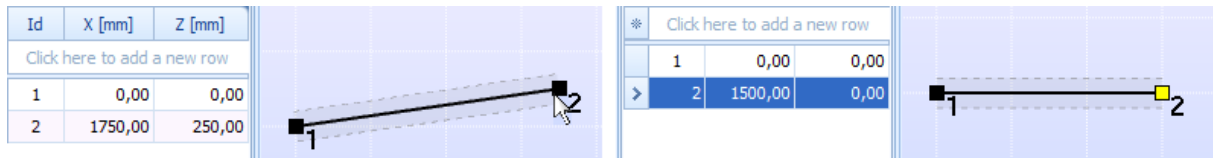
Funksjonaliteten *snap to grid* brukes for å kunne plassere ut og flytte *joints* og *members* hurtig og presist. I versjon **3.0** kunne det dessverre skje at *joints* *snappet* til *grid* når man bare klikket på dem, ikke flyttet dem.

Årsaken til feilen er enkelt fortalt at hver gang en museknapp ble trykket tolket programmet det som om en flytteoperasjon ble startet. Derfor forsøkte *snap to grid*-funksjonen å finne et *snap*-punkt for objektet som ble markert.

Grunnen til feilen er enkelt fortalt at hver gang musen trykkes ned tolker PC-en det som om musen både blir trykket ned og flyttet. Derfor prøver *snap to grid*-funksjonen å finne ut hvor det nærmeste hjørnepunktet er, fordi den alltid vil prøve å relassere et flyttende objekt til rutenettet.

Løsningen på problemet ble å legge inn et kriterium som registrerer om musen har blitt flyttet langt nok langs skjermen til at *snap to grid* skal aktiveres. Denne grensen er lagt til 28 piksler. Undervegs i utviklingsarbeidet brukte man et annet kriterium, som var angitt i *world coordinates*. Dette ble skiftet ut fordi det ikke virket hensiktsmessig når det ble zoomet langt ut. En mer omfattende beskrivelse finnes i vedlegg A.3.1.





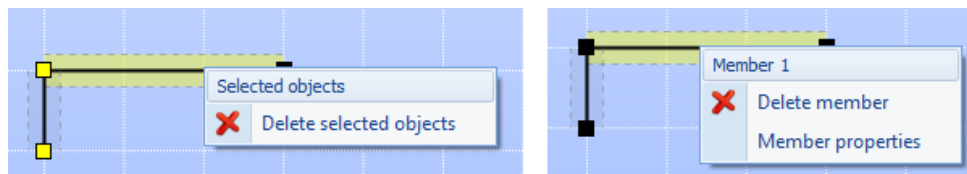
Figur 3.36: I versjon 3.0 kunne det oppstå feilaktig *snapping* når man trykket på *joints*.

### Mistanke om feil i *undo/redo*

Ved ett tilfelle ble det rapportert at *undo* gikk to steg tilbake i modelleringsprosessen. Tilsynelatende ble både flyttingen av et *joint* og fjerningen av en *rotational spring* angret i én og samme *undo*-operasjon. Det viste seg at opphavet til feilen faktisk var *snap* ved museklikk, ikke *undo/redo*.

### 3.3.5 Feilaktig markering av objekter vha. høyretasten

I versjon 3.0 var det mulig å markere flere grafiske objekter samtidig, ved å trykke på dem i rekkefølge med høyre museknapp. Det burde ikke gått an, med mindre CTRL-tasten ble holdt nede. Denne enkle glippen ble fort ordnet. Figur 3.37 illustrerer endringen.



Figur 3.37: Versjon 3.0 (t.v): Mulig å markere flere objekter m/høyretasten -  
Versjon 3.1 (t.h): Ikke lenger mulig, med mindre CTRL holdes nede.

### Feil i *selection filter*

Dersom et annet filter enn *all* er brukt i *selection filter*-funksjonen fra *toolbox panel* skal det bare være mulig å markere visse objekter, f.eks. bare *joints*. Når man brukte høyretasten til å markere objekter var det ikke tatt høyde for filteret, så alle objekter kunne markeres uavhengig av valgt filter. Oppførselen er rettet.

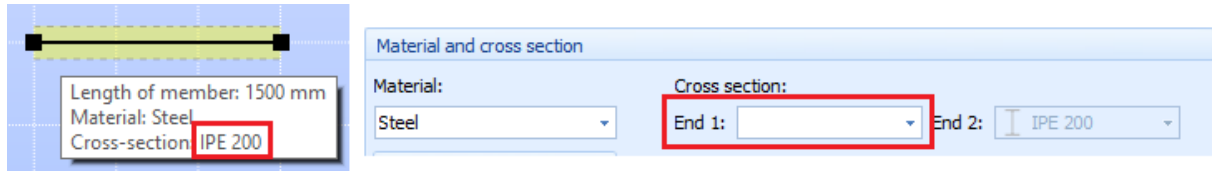
## 3.4 Konstruksjonsmodell

En konstruksjonsmodell er en matematisk modell av en reell konstruksjon, som beskriver geometri, materiale, randbetingelser og belastning på en idealisert måte. Denne modellen er utgangspunktet for den diskretiserte beregningsmodellen og det er derfor viktig at modellen er korrekt. Endringene i dette avsnittet er utført for å forsøke å oppnå dette. Forandringene er detaljert beskrevet i vedlegg A.4.

### 3.4.1 Mindre endringer i konstruksjonsmodell

#### Feilrettinger

Tverrsnitt som er i bruk kan ikke lenger slettes, slik at man unngår at referanser forsvinner, eller at blanke felter vises i nedtrekkslister for valg av tverrsnitt.



Figur 3.38: Versjon 3.0: Tvetydighet i *member properties dialog* pga. sletting av tverrsnittet “IPE 200”.

Verdien av rotasjonstivheten til rand- eller koblingsfjær ble endret med en faktor  $57.32 = \frac{180}{\pi}$  hver gang man åpnet dialogboksen som viser dens egenskaper. Feilen er rettet i vedlegg A.4.17.

Rotasjonsfjærer skaleres ikke lenger med feil enhetskonverteringsfaktor. Faktoren var satt til å konvertere aksialstivhet istedenfor rotasjonsstivhet.

Ved endring av koblingsfjærer oppstod en `NullReferenceException` fordi man prøvde å operere på dataobjekter som var satt til null, noe som ikke gir mening for en datamaskin. Feilen er omtalt i A.4.16. Behandling av slike *exceptions* har vært viktig for å unngå programstopp.

#### Modifisering

Når *member properties dialog* åpnes for mer enn en *member* vil de ofte ha ulike egenskaper, eksempelvis ulike tverrsnitt. For å reflektere dette vises teksten “<multiple>” som plassholder i tekstfelt (eller nedtrekksliste). Dersom en enkelt *submember* har interpolert tverrsnitt vil “<interpolated>” vises i nedtrekkslisten for tverrsnitt.

### 3.4.2 Member-retning

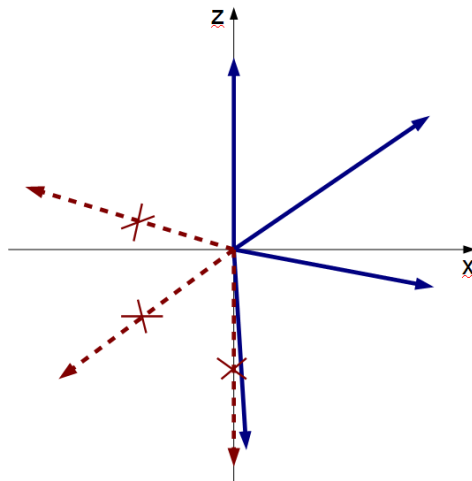
*Members* skal være orientert fra venstre mot høyre (V-H), ikke i tilfeldig retning, som illustrert i figur 3.39. Årsaken til dette er at intuitiv visning av egenskapene til *members* forutsetter V-H-orientering. I vedlegg A.4.14 *Member-retning* er det gitt konkrete eksempler på at tilfeldig retning kunne skape forvirring i forrige versjon.

For å unngå å begrense modelleringsprosessen er det bestemt at *members* fremdeles kan utplasseres med vilkårlig retning. Programmet skifter nå automatisk retning på *members* i det bruker velger å utføre analyse, før *meshing* gjennomføres. Dette sikrer at bruker ikke merker endringen mens modellering foregår. Det er bestemt at bruker ikke trenger å få beskjed om at endringen er gjort fordi selve geometrien til modellen er uendret.

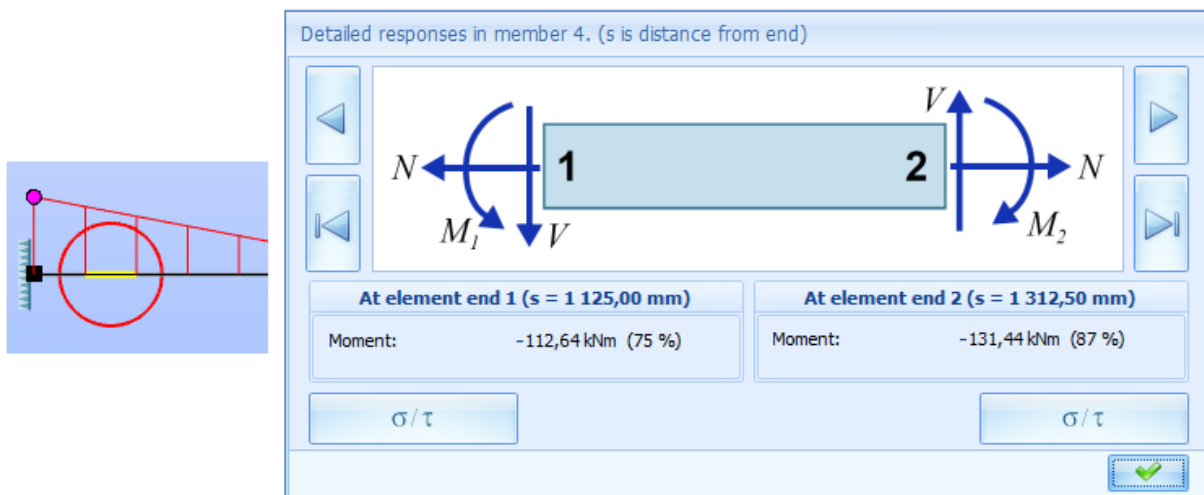
Følgende endringer kan observeres av bruker:

- Høyden/radius av *arch members* bytter fortegn.
- Tverrsnittene bytter side i dialogboksen for *member properties dialog* slik at det er samsvar mellom modellvinduet og dialogboksen.
- Nodene bytter side i dialoger for elementresultater slik at det er samsvar mellom resultatvinduet og dialogboksen. Figur 3.40 og 3.41 illustrerer dette.

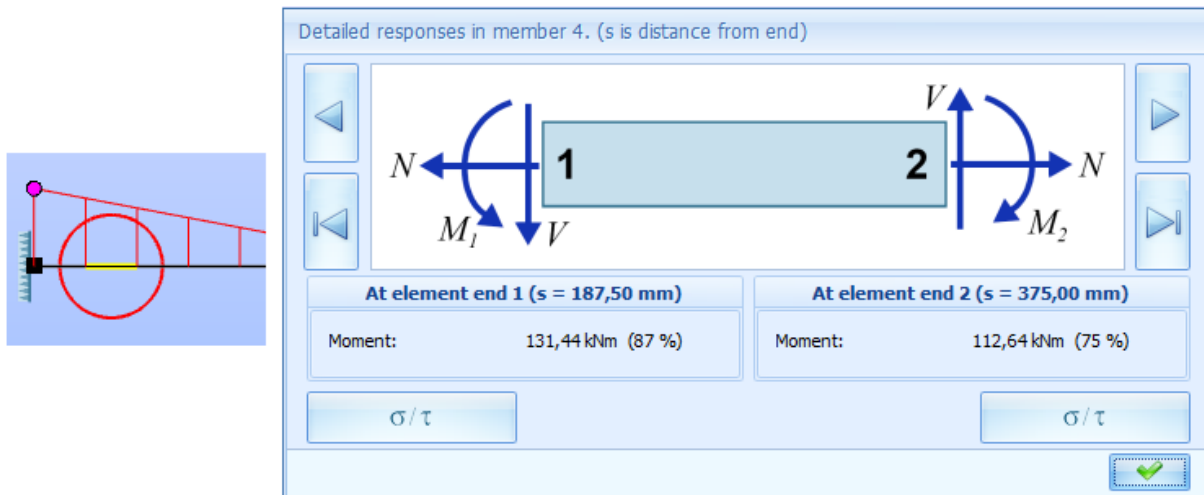
Betingelsen for å bytte retning er at *joint*-et i ende 2 er til venstre for *joint*-et i ende 1. Dersom begge *joints* er på samme x-koordinat skifter *member*-en retning hvis *joint*-et i ende 2 er under *joint*-et i ende 1.



Figur 3.39: Diagram for gyldig retning av *members*, der ende 1 er i lokalt origo. *Members* vist med stiplet linje blir tvunget til å skifte retning.



Figur 3.40: Versjon 3.0: *Member* tegnet fra høyre mot venstre - Momentverdier i feil rekkefølge siden elementene også blir definert fra høyre mot venstre. (Dialogen er forenklet for å trekke frem essensen).

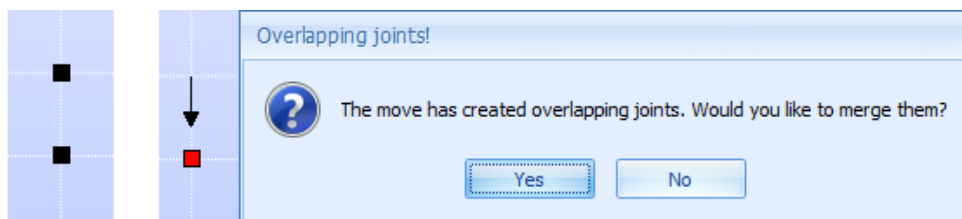


Figur 3.41: Versjon 3.1: *Member* tegnet fra høyre mot venstre - Momentverdier i riktig rekkefølge siden elementene defineres fra venstre mot høyre. (Dialogen er forenklet for å trekke frem essensen).

### 3.4.3 Overlappende *structural items*

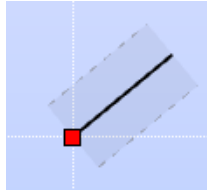
I versjon 3.0 ble det implementert logikk for å markere *joints* som overlapper med rødfarge, slik at brukeren ser at det er flere i samme punkt. [9, avsnitt 5.1] Dette var viktig å implementere fordi man kunne skape statisk ubestemte modeller uten å være klar over det.

I versjon 3.1 er det bygget videre på arbeidet til Pedersen [9] ved å gi mulighet for å slå sammen overlappende *joints*. Dersom det skapes slike *joints* varsles brukeren vha. en dialogboks og får valget mellom å slette ett av *joint*-ene eller å beholde begge to. Om de beholdes vil de markeres med rødt helt til ett av dem evt. blir flyttet bort.



Figur 3.42: Versjon 3.1: To separate *joints* (t.v.) - Flytting fører til overlapping, og bruker velger om *joint*-ene skal slås sammen (t.h.).

Ett av tilfellene der det er gunstig å slå sammen *joints* er når man flytter det ene *joint*-et i en *member* over det andre, slik at *member*-en ikke får utstrekning. I slike tilfeller slettes både ett av *joint*-ene og selve *member*-en, med mindre bruker svarer "nei" på spørsmålet i figur 3.42.



Figur 3.43: Versjon 3.0: Mulig å lage *member* med overlappende *joints*.

Av erfaring merker man at det sjelden er behov for overlappende *joints* i en modell. Derfor er det lagt til en innstilling i *settings*<sup>4</sup> som gjør at *joints* alltid blir slått sammen. Det anbefales å lese vedlegg A.4.11 og A.4.10 om man vil vite detaljene.

*Joints* kan altså fremdeles være overlappende, noe som tillater at *members* også kan overlappe. Et problem der flere *members* ble slettet samtidig, spesielt hvis noen av dem overlappet, er nå håndtert ved å akseptere at rekkefølgen som *joints* slettes i kan være tilfeldig. Rettingen er mer utdypet i vedlegg A.4.13.

Effektene som oppstår ved bruk av overlappende *structural items* diskuteres mer i 4.3 “3D-effekter” i **fap2D**.

### 3.4.4 Arch members

For å lage *members* som krummer nedover kunne man modellere med negativ høyde eller radius i versjon 3.0. Programmet opprettet stadig *unhandled exceptions* forbundet med denne typen *members*, derfor ble det bestemt å fjerne denne praksisen. Alternative modelleringsmetoder var nødvendig for å modellere tilsvarende *arch members*<sup>5</sup>.

Negativ høyde eller radius ble senere gjeninnført blant annet fordi deltakerne i betatesten syntes praksisen var mer intuitiv enn det foreslåtte alternativet.<sup>6</sup> Det var også nødvendig for å gjennomføre endringen med å orientere *members* fra venstre mot høyre (se kapittel 3.4.2 *Member-retning*). Retting av andre feil forbundet med utplassering av *members* gjorde programmet robust nok til å takle gjeninnføringen. Se vedlegg A.4.4 Arch member med negativ radius for detaljer.

For å hindre at *parabolic arch members* beregnes med urealistisk geometri vises nå advarsler dersom forholdet mellom pilhøyde og kordelengde blir for stort. Dersom forholdet blir større enn 3 vises en advarsel til brukeren, og dersom forholdet blir 10 eller høyere blir brukeren hindret fra å gjennomføre modelleringen. **fap2D** bør nemlig ikke brukes til å analysere slike krappe konstruksjoner siden bruddmekanisk teori sannsynligvis vil bli dimensjonerende.

Error List		
Error...	Description	Time stamp
	The geometry of one or more parabolic arch members will cause numerical instabilities ( $h/L > 10$ ). The joint will not be moved.	14:33:02
	The geometry of the arch member may cause numerical instabilities! ( $h/L > 3$ )	14:31:39

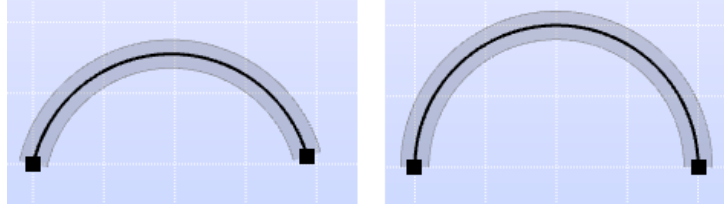
Figur 3.44: Advarsler i *error list* pga. krappe/spisse *parabolic arch members*.

<sup>4</sup>Innstillingen presenteres i vedlegg A.9.9.

<sup>5</sup>Man kunne modellere en *arch member* som krummet nedover ved å trekke *member*-en fra høyre mot venstre. Det lokale aksesystemet ble da rotert 180°.

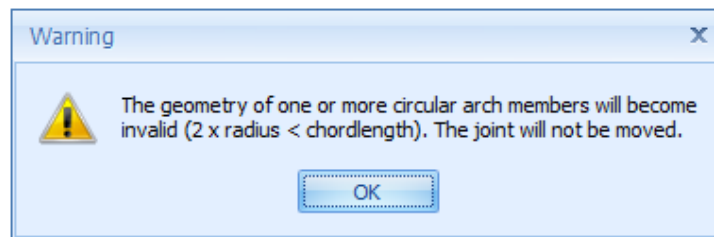
<sup>6</sup>Se kapittel 4.1 Betatesting av **fap2D 3.0.0 Beta**.

Bruker blir også hindret fra å modellere *circular arch members* med ugyldig geometri. Hensikten med dette er først og fremst å unngå programstopp. Dersom kordelengden blir større enn  $2 \cdot \text{radius}$  vises en advarsel i *error list* og *arch memberen* blir tilbakestilt til sin opprinnelige geometri. En tilsvarende advarsel finnes også i *member properties dialog*. Detaljer kan finnes i vedlegg A.4.1 Validering av *arch member*-geometri.



Figur 3.45: Versjon **3.0** (t.v.): Ikke mulig å *snappe* til halvsirkel -  
Versjon **3.1** (t.h.): Full halvsirkel er mulig.

Validering for *circular arch members* er korrigert slik at det kan lages komplette halvsirkler. I tillegg er inndeling av *arch members* langs kurven rettet slik at det deles i like lange *submembers*.

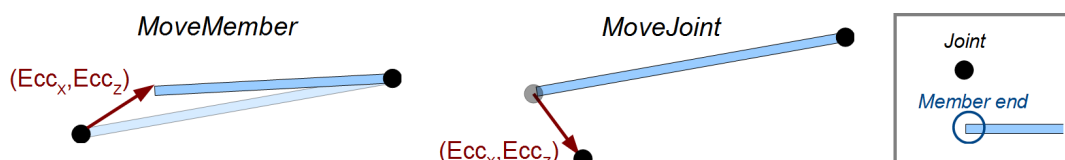


Figur 3.46: Dialog med varsel om at radius i en *circular arch member* er for liten.

### 3.4.5 Eksentrisiteter

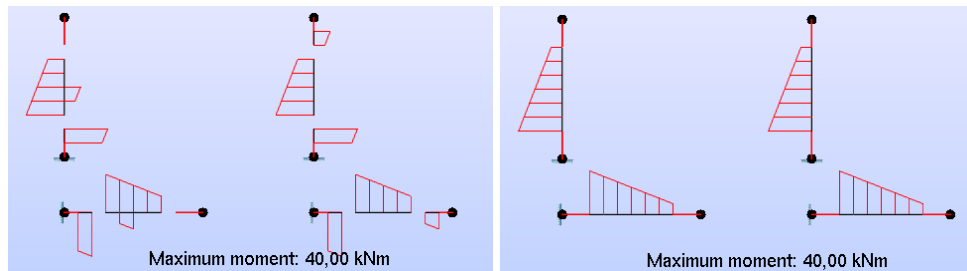
Eksentrisiteter påvirker plasseringen av *joints* og geometrien til *members*. Før en eksentrisitet settes på modellen blir alle tilknyttede *members* sjekket for gyldig geometri i hht. validering av *arch members* nevnt tidligere i avsnitt 3.4.4 *Arch members*.

To ulike typer eksentrisitet er definert, *MoveMember* og *MoveJoint*, som hhv. flytter enden av en *member* i forhold til sitt tilknyttede *joint*, eller som flytter et *joint* i forhold til én eller flere tilknyttede *members*



Figur 3.47: Definisjon av de to typene eksentrisiteter: *MoveMember* og *MoveJoint*.

Forskjellene mellom de to eksentrisitetstypene ble inkonsekvent håndtert. Elementer ble malplassert og rotert vekk fra sin virkelige posisjon når de skulle tegnes i resultatdiagrammene. Beregning av posisjonene til *members* og elementer med eksentrisiteter er nå refaktorisert slik at det skilles mellom posisjonen til *member*-ende og *joint* i hht. figur 3.47.

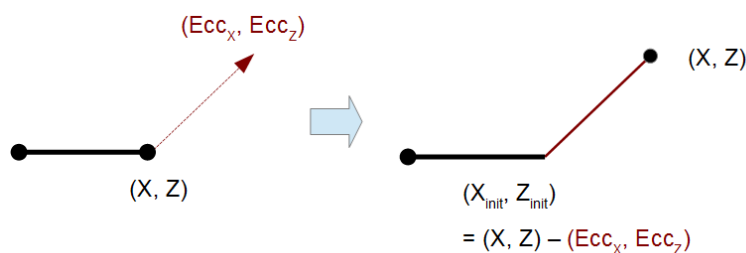


Figur 3.48: Eksempel på momentdiagram hhv. før (t.v.) og etter ny håndtering av de ulike eksentrisitetstypene (t.h.). Begge eksemplene inneholder eksentrisiteter av typen *MoveMember* til venstre i modellen, og *MoveJoint* til høyre i modellen.

Definisjonen av eksentrisitetstypen *MoveJoint* er endret slik at *joints* kan håndtere flere samtidig. Forflytningen av *joint*-et blir beregnet fra summen av vektorene for hver eksentrisitet. Selve *joint*-et blir flyttet slik at tilknyttede *structural items* kun trenger å forholde seg til den nye *joint*-posisjonen, ikke eksentrisitetene. Den originale plasseringen av *joint*-et beregnes ved hjelp av eksentrisitetene. Mer utfyllende beskrivelse kan finnes i vedlegg A.4.9 Eksentrisiteter ved flere *MoveJoints*.



Figur 3.49: Et *joint* med to eksentrisiteter av typen *MoveJoint*.



Figur 3.50: Ny definisjon av *joint*-koordinater med eksentrisiteter: *Joint*-koord. og eksentrisitet ved opprettelse av eksentrisitet (t.v.) - *Joint*-koord. og eksentrisitet etter opprettelse (t.h.).

Tegningen av *MoveJoint* i *user control* for eksentrisiteter ble fikset så den kan håndtere flere *MoveJoints*. Et annet problem forbundet med *MoveJoint*, som nå er reparert, er at *joints* ikke ble flyttet tilbake til sine opprinnelige posisjoner da de ble slettet.

## 3.5 Beregningsmodell og analyser

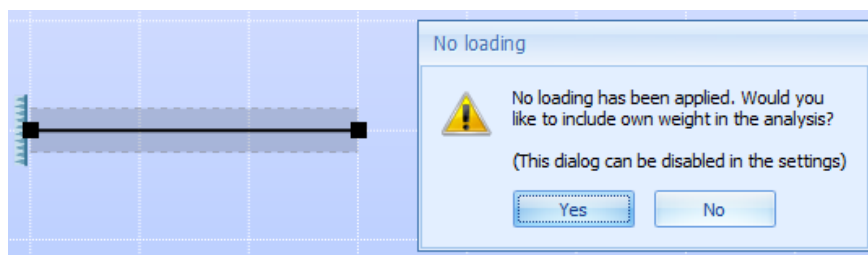
Dette avsnittet dreier seg om opprettelse av beregningsmodellen og konfigurasjon av oppsett for ulike analysetyper.

### 3.5.1 Standardverdier for *load combinations* og *time axes*

For å utføre de fleste analyser i **fap2D** må det defineres en aktiv *load combination* (lastkombinasjon) slik at konstruksjonen blir påkjent av laster. For enkelte av analysene i versjon **3.0** var det oppgitt en *default load combination* som var aktiv selv om brukeren var klar over det. Andre analyser hadde ingen *default* og dette gjorde at det var dårlig samsvar på tvers av analyser. Blant annet hadde lineær statisk analyse en *default load combination* som la til egenvekt når ingen eksterne laster ble satt på. Dette kunne forvirre brukeren.

I versjon **3.1** er det gjort endringer basert på følgende argumentasjon:

- Alle *default*-verdier må være synlige for brukeren, hvis ikke skal de fjernes. Dette medfører at egenvekt ikke lenger blir lagt til automatisk.
- Dersom det ikke er angitt en aktiv *load combination* (eller det ikke finnes laster i den) skal brukeren varsles via en dialogboks når det forsøkes å utføre en analyse. Dialogboksen skal gi brukeren mulighet til å gå til analysens *loading dialog*, der lastkombinasjon kan angis.
- For dynamiske analyser der det brukes tidsakser skal brukeren i tillegg varsles dersom tidsaksen ikke er angitt korrekt, og kunne endre den med en gang.



Figur 3.51: Dialogboks som vises når det ikke er lagt til ytre laster i lineær statisk analyse. Tilsvarende figurer finnes for andre analysetyper.

Endringene som er innført gjør **fap2D** mer konsekvent og brukervennlig siden brukeren slipper å tolke mange ulike feilmeldinger som oppstod pga. *load combinations* og *time axes*. Det fokuseres altså på å guide brukeren til å raskere løse årsaker til feilmeldinger, ikke bare påpeke dem. Forhåpentligvis vil det gjøre at brukeren raskt blir klar over at noen av knappene i *analysis ribbon page* alltid må besøkes før en analyse utføres.

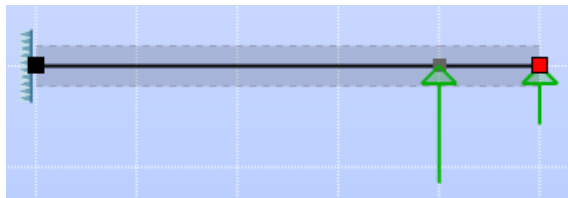
Vedlegg A.5.1 tar for seg en feilmelding for *buckling analysis* som var vanskelig å tolke og dermed satte brukeren ut av spill.



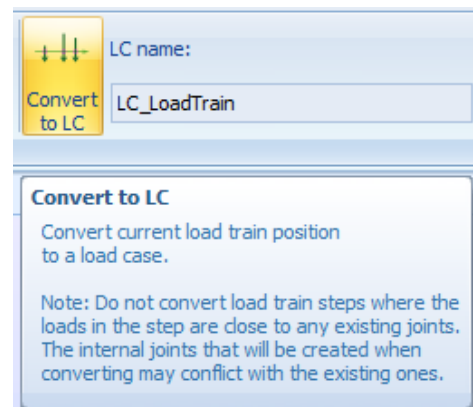
### 3.5.2 Konvertering av laststeg til statisk lasttilfelle

I *real time* og *influence line analyses* er det mulig å konvertere et laststeg fra analysene til et statisk lasttilfelle. Funksjonaliteten som skulle konvertere laststeget opprettet *internal joints* for å putte lastene i, men tok ikke hensyn til de eksisterende *joints* i modellen og det kunne resultere i en ugyldig konstruksjonsmodell og i verste fall programstopp.

Feilen nevnes i dette kapitlet fordi konverteringen blir gjort under *analysis ribbon page*. Den er ikke rettet i versjon 3.0, men det er innført flere tiltak som skal hindre brukeren fra å opprette ugyldige lasttilfeller. Ytterligere detaljer finnes i vedlegg A.5.6.

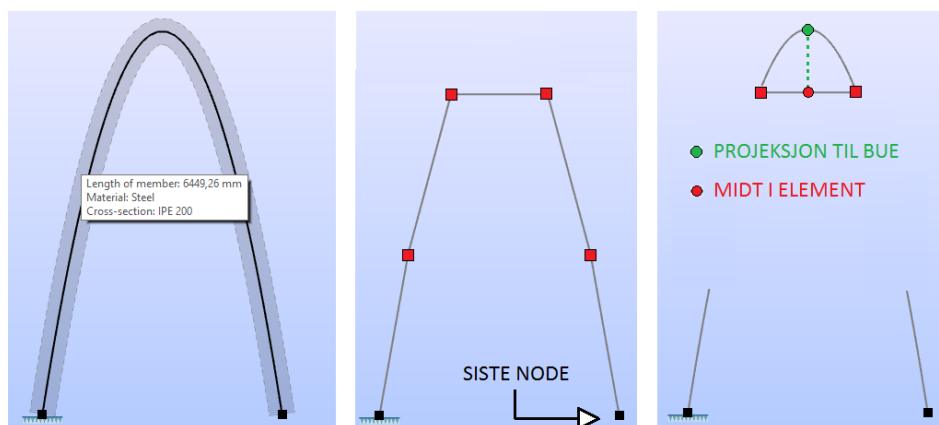


Figur 3.52: Dersom man prøver å konvertere et lasttog (bestående av to laster) for utrageren i figuren vil det interne *joint*-et som opprettes t.h. overlappe med et eksisterende *joint*.



Figur 3.53: *Tooltip* er brukt som tiltak for å hindre brukeren i å lage ugyldige modeller.

### 3.5.3 Unøyaktig beregning av nodeplassering langs *arch members*



Figur 3.54: *Arch member* som deles i fem elementer (bilde t.v. og i midten). Elementlengden mellom nodene i øverste element er liten i forhold til buelengden mellom disse nodene (bilde t.h.). Bildet t.h. viser hvordan midtpunktet i et element er tvetydig definert i **fa2D**. Avstanden fra 1. (venstre) *joint* til nodene i elementet ble i versjon 3.0 beregnet feil: Avstand = Buelengde (1. *joint* til buetopp (grønn sirkel))  $\pm \frac{\text{elementlengde}}{2}$ . I versjon 3.1 benyttes bare buelengder, ikke en blanding mellom bue- og elementlengde.

Plasseringen av noder langs *arch members* beregnes slik at buelengden mellom nodene er lik. Nodekoordinatene beregnes riktig når de plasseres ut, men når det senere skal refereres til dem ved *lumping* av linjelaster og presentasjon av elementresultater blir avstanden fra 1. *joint* i *member*-en og fram til nodene beregnet feil. Figur 3.54 forsøker å vise dette.

Oppførselen ble oppdaget sent i masteroppgaven og man har ikke fått løst problemet fullstendig. Likevel konkluderes det med at avstandsberegningene i praksis er riktige fordi man bør bruke en fin elementinndeling for å modellere en reell *arch member*. Det er gjort enkel validering i avsnitt 4.5.2 som underbygger denne antagelsen.

### 3.5.4 Nedre grense for elementlengde

#### Teoretisk bakgrunn

Antall elementer langs *members* og minste elementlengde har betydning for hvor korrekte resultatene fra elementanalysen blir. **fap2D** håndterer feil som kan oppstå f.eks. på grunn av dårlig *kondisjonert* stivhetsmatrise ved hjelp av *test av diagonal nedbrytning* (*diagonal decay test*)<sup>7</sup>, heretter referert til som DNT.

DNT er en enkel og beregningsmessig billig test for å oppdage numeriske feil i ligningsløseren. Det aktuelle ligningssystemet er stivhetsrelasjonen  $\mathbf{K}\mathbf{r} = \mathbf{R}$  der  $\mathbf{K}$  faktoriseres<sup>8</sup> til

$$\mathbf{K} = \mathbf{LDL}^T \quad (3.3)$$

Testen sammenligner forholdet mellom diagonalelementene  $D_{ii}$  i  $\mathbf{D}$  med tilsvarende  $K_{ii}$  i  $\mathbf{K}$ . DNT antyder at systemet er singulært dersom

$$|D_{ii}| < |\epsilon \cdot K_{ii}| \quad , \quad \epsilon = 10^{-r} \quad (3.4)$$

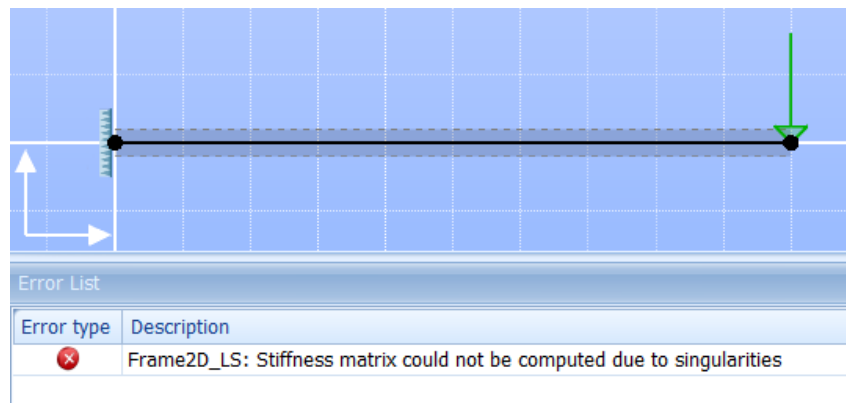
Konstanten  $r$  er et positivt heltall som indikerer at  $r$  signifikante sifre er "mistet". I **fap2D** er størrelsen for testen valgt til  $\epsilon = 10^{-8}$ . Denne ble justert fra  $\epsilon = 10^{-10}$  etter at et fysisk singulært system ble feilaktig godkjent av testen.

#### Observasjoner ved ekstrem elementinndeling

Det er valgt å gi brukeren en advarsel dersom elementinndelingen blir liten fordi det ikke er sikkert at numeriske feil oppdages av DNT. Dette kan observeres i den innspente bjelken i figur 3.55 der elementene er nummerert fra den innspente enden og ut til den fri. Når det nummereres på denne måten vil testen slå ut for så få som 465 elementer i **fap2D**, uavhengig av valgt lengde ( $L = 1\text{m}, 10\text{m}, 100\text{m}$ ) og tverrsnitt (IPE200 og Rec60x100). Derimot vil testen akseptere samme modell dersom nummereringen er speilet slik at den starter i den frie enden og går til den innspente enden.

<sup>7</sup>Se s. 243-247 i [4] og avsnitt 9.2-9.4 i [1]

<sup>8</sup> $LDL^T$  faktorisering, også kalt LDL faktorisering eller LDLT faktorisering, er en variasjon av Gauss eliminasjon.



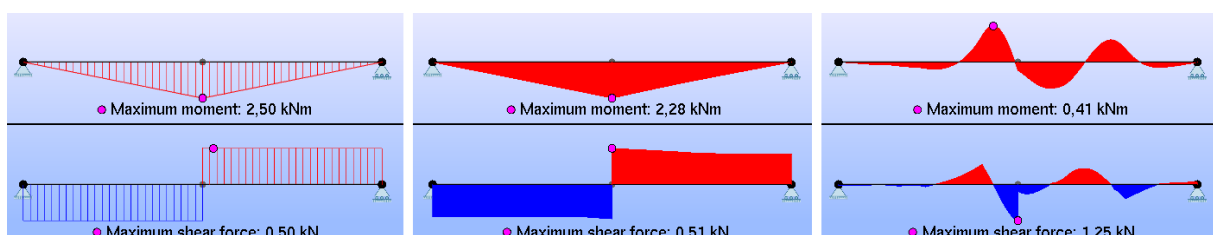
Figur 3.55: Fast innspent bjelke med 500 elementer, nummerert fra innspent til fri ende.

Analyser gjennomført med modeller som slipper igjennom DNT forventes å gi korrekte resultater. Enkle forsøk viser at den innspente bjelken med nummerering fra den fri enden beregnes rimelig korrekt med flere tusen elementer. Et så stort antall elementer for kun én *member* er strengt tatt unødvendig. Dersom man regnet på den fast innspente bjelken med punktlast i figur 3.55 ville kun ett bjelkeelement vært tilstrekkelig for å gi eksakt løsning. Det er likevel interessant å se hva som skjer dersom antall elementer økes. Figur 3.56 viser moment- og skjærkraftdiagrammer for en fast innspent bjelke med hhv. 50, 5000 og 100000 elementer. Ved 5000 elementer kan man merke små endringer i skjærkraft- og momentdiagrammene. Ved 10000 eller flere blir resultatene betydelig feil. Store feil merkes derimot ikke før ca 10000 elementer for en fritt opplagt bjelke. Se figur 3.57.

Som nevnt er flere tusen elementer for én *member* allerede unødvendig mange, og 100000 elementer er en helt ekstrem inndeling. Ved vanlig bruk av **fap2D** er en slik elementinndeling uaktuell, og som vist er det også risikabelt siden DNT ikke alltid oppdager slike numeriske problemer.

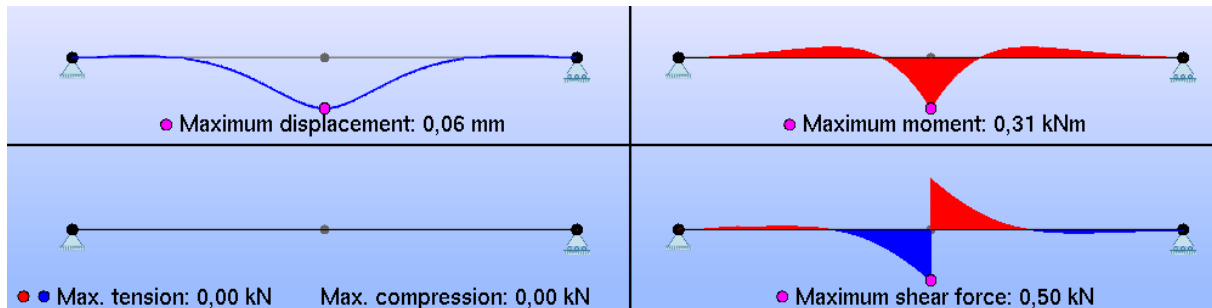


Figur 3.56: Moment- og skjærkraftdiagram for fast innspent bjelke med hhv. 50, 5000 og 100000 elementer.



Figur 3.57: Moment- og skjærkraftdiagram for fritt opplagt bjelke med hhv. 50, 10000 og 100000 elementer.

Det bør nevnes at de numeriske feilene kan bevare symmetrien i problemet for enkelte elementinndelinger. Dette kan observeres for den fritt opplagte bjelken med 80000 elementer i figur 3.58. Årsaken til at symmetrien bevares for enkelte tilfeller men ikke for andre er det vanskelig å si noe om.



Figur 3.58: Reaksjoner for fritt opplagt bjelke med 80000 elementer med symmetriske numeriske feil.

### Årsak til numeriske feil pga. elementinndelingen

Årsaken til de observerte numeriske feilene er en blanding av fysiske forhold og regneunøyaktigheter (bla. *truncation*<sup>9</sup>) på grunn av begrenset presisjon. Størrelsesforskjellen på små deformasjoner og store stivlegemebevegelser er en klassisk fysisk forklaring på numeriske problemer. Elementene i **fap2D** er bjelkeelementer der stivheten er gitt av forholdet mellom andre arealmoment (eller areal) og en potens av elementlengden, f.eks.  $K_{11} = \frac{12EI}{L^3}$ . Når elementlengden  $L$  går mot 0 vil stivheten øke betraktelig. Den totale forskyvningen av elementet beregnes både av deformasjonen av elementet selv og stivlegemebevegelser. En forklaring er derfor at for stive elementer med små deformasjoner og store forskyvninger på grunn av stivlegemebevegelser vil deformasjonene “forsvinne” sammenlignet med stivlegemebevegelserne.

Alle relevante desimaltall eller flyttall i **fap2D** er representert med dobbel flyttallspresisjon ved bruk av datatypen `double`. Både C# og FORTRAN representerer `double` nøyaktig til 15-16 signifikante sifre. Som vist er det likevel mulig at denne typen presisjon ikke er tilstrekkelig for å representere elementforskyvningene etter mange operasjoner i ligningsløseren. Samtidig er elementinndelingene som det opereres med i dette delkapittelet langt utenfor vanlig bruksområde, og av mer akademisk interesse.

### Innføring av advarsel for elementinndeling

For å unngå unødig fine og potensielt problematiske elementinndelinger, eller kanskje mest for å unngå ujevn deling når en har en blanding av korte og lange *members* er det valgt å la programmet vise en dialogboks med advarsel dersom brukeren oppretter elementer som er kortere enn 5% av høyden til tilhørende tverrsnitt. Brukere med mye erfaring kan skjule denne advarselen i *global settings*.

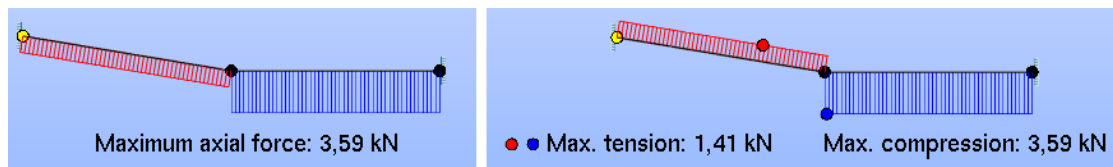
<sup>9</sup>*Truncation*: forkasting av et visst antall av de minst signifikante sifrene.

## 3.6 Presentasjon av resultater

Dette kapittelet tar for seg forbedringer av resultatpresentasjonen. Hensikten med dem er å gjøre resultatene mer innlysende og informative. Endringene er dokumenterte i vedlegg A.6.

### 3.6.1 Strekk- og trykkrefter

Aksialkrefter kan opptre enten som strekk eller trykk og i **fab2D** har kreftene hhv. positivt og negativt fortegn. Da den maksimale aksialkraften ble betegnet i versjon **3.0** var det i form av den største absolutte aksialkraften. Det ble ikke påpekt om dette var i form av strekk eller trykk, så i versjon **3.1** vises både den maksimale positive og negative aksialkraften. Dette er mer informativt for brukeren. De maksimale kreftene markeres med hhv. rødt og blått symbol i diagrammet slik at man også ser hvor de oppstår. Mer om dette i avsnitt 3.6.2.



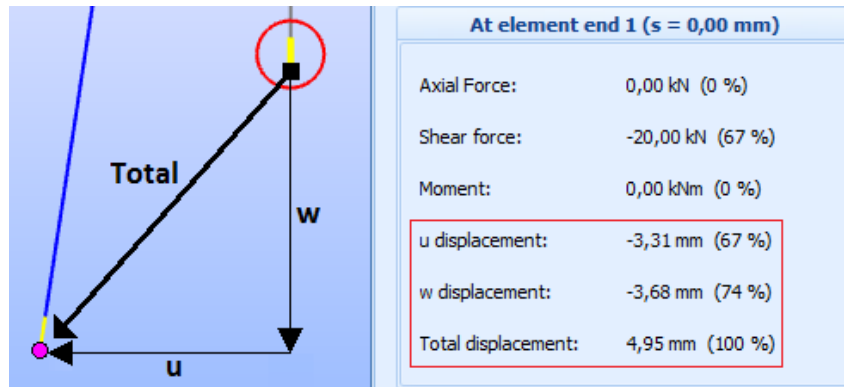
Figur 3.59: Aksialkraftdiagram fra versjon **3.0** (t.v) og versjon **3.1** (t.h).

### 3.6.2 Markering av maksimalrespons i resultatdiagrammer

Det er lagt til et symbol som indikerer hvor den maksimale responsen i resultatdiagrammer oppstår, slik at det er lettere å se hvor belastningen er størst. Symbolet brukes i resultatdiagrammene som viser forskyvning, bøyemoment, aksialkraft, skjærkrefter og utnyttelse av kapasitet i hht. ståldimensjonering. Bildet t.v. i figur 3.59 illustrerer bruken av symbolet.

### 3.6.3 Maksimal forskyvningsvektor

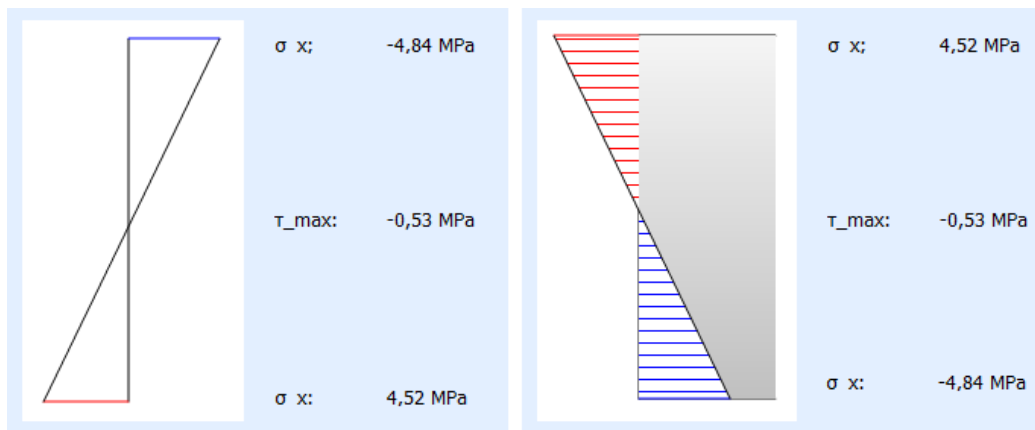
Presentasjonen av elementresultatene er forbedret ved at forskyvninger beregnes både komponentvis og som total vektorforskyvning. Maksimal forskyvning angis nå som den maksimale verdien blant de totale forskyvningene istedenfor den maksimale blant alle komponentverdiene. Elementresultatene kan ses ved å høyreklikke på et element for å åpne *section forces dialog*.



Figur 3.60: Versjon 3.1: Total forskyvningsvektor inkluderes i resultat-presentasjonen, sammen med komponentene (som ble vist fra før).

### 3.6.4 Forbedret visning av spenninger

Visningen av spenningene i elementene er forbedret med korrekt presentasjon av fordelingen av strekk- og trykkspenninger over tverrsnittshøyden. I versjon 3.0 tok ikke figuren som skulle vise fordelingen høyde for om trykk- og strekkspenningene var i overkant eller underkant av tverrsnittet. Det var tilsynelatende ikke samsvar mellom belastningen som ble vist i modellvinduet og i dialogen med spenninger. Strekkspenninger ble (som vist i figur 3.61) alltid tegnet til venstre og trykk til høyre, uavhengig om man så på spenningene i venstre eller høyre ende av et element.



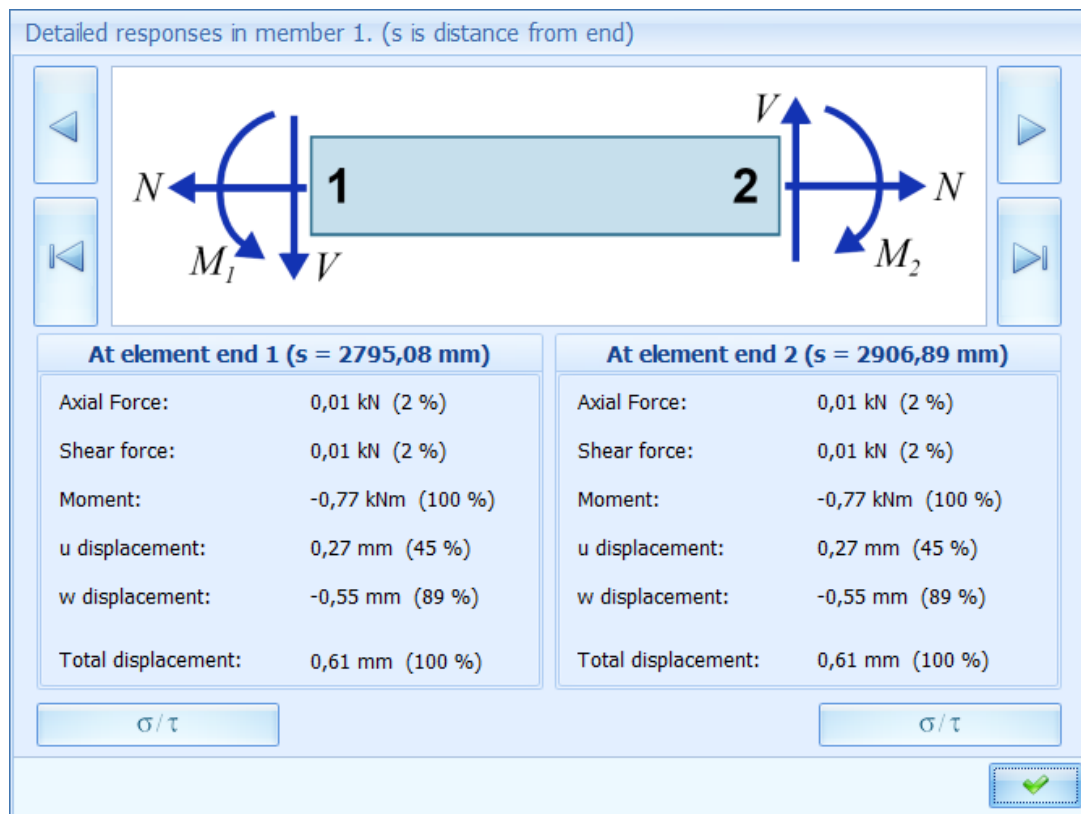
Figur 3.61: Visning av spenninger i hhv. versjon 3.0 (t.v) og 3.1 (t.h). I bildet til høyre er enden av elementet med i visualiseringen og strekkreftene tegnes slik at de “strekker” elementet, mens trykkspenningene “trykker sammen” elementet.

### 3.6.5 Presentasjon og navigering av elementresultater

I resultatvisningene er det mulig å høyreklikke på et element for å vise frem en dialogboks med elementkrefter og -forskyvninger. For at man skal slippe å gå ut av dialogboksen for å se responsen i et annet element har det blitt innført navigeringsknapper i dialogen i versjon 3.1. Navigeringsknappene vises i figur 3.62.

Navigeringen foregår begge veier, enten ved hjelp av de nevnte knappene eller ved å bruke piltastene på tastaturet. Siden elementene er knyttet til *members* kan man kun navigere mellom elementer i samme *member* (inkludert alle *submembers*). Det er valgt å begrense navigeringen slik fordi det er vanskeligere å definere hvordan man skal hoppe mellom ulike *members*. Dessuten tar det ikke så lang tid å gå ut av dialogen for å gå inn i en ny en.

Implementasjonen ble laget pga. at enkelte syntes, under betatesting, at det var tungvint å bla mellom elementer.



Figur 3.62: Versjon 3.1: Dialogboks for elementresultater med knapper for navigering mellom elementer i samme *member*.

### 3.7 Steel design

Larsen [5] utviklet i 2011 dimensjoneringskontroller<sup>10</sup> for stålkonstruksjoner i hht. Norsk standard [2]. Selve dimensjoneringskontrollene fungerer tilsynelatende godt, men det har blitt utført en del endringer i visualiseringen av resultater. I tillegg har det blitt lagt til mer informasjon, og noen av kontrollene er avgrenset slik at de ikke benyttes på konstruksjoner de ikke er ment for.

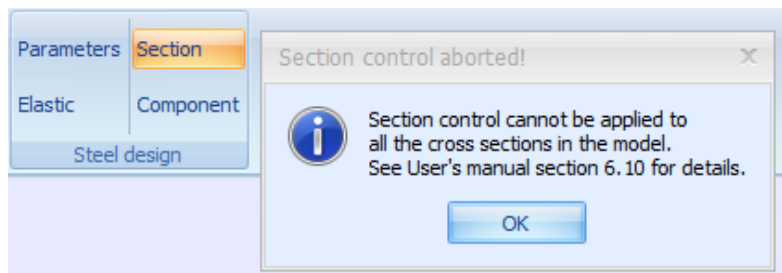
Endringene er såpass omfattende at det er valgt å presentere dem i et eget kapittel. Det er verdt å merke seg at informasjonen om *steel design* i vedlegg A *Release notes* - Versjonsmerknader ikke er helt komplett.

<sup>10</sup>Kontroller i hht. ståldimensjonering må ikke forveksles med *controls* i GUI.

## Avgrensning av kontroller

Larsen har selv rapportert hvilke avgrensninger som bør gjelde for ståldimensjoneringskontrollene. Det er av ulike grunner bare mulig å utføre kontroller for bestemte typer tverrsnitt og *members* [5, avsnitt 6.1]. Hvilke tverrsnitt som kan benyttes varierer avhengig av om kontrollen som kjøres er *section*, *elastic* eller *component*. Det var mulig å benytte *steel design* for dynamiske analyser i versjon **3.0**, men det ble bestemt at dette måtte endres i versjon **3.1** siden teorien først og fremst er beregnet for lineær og ikke-lineær statisk analyse.

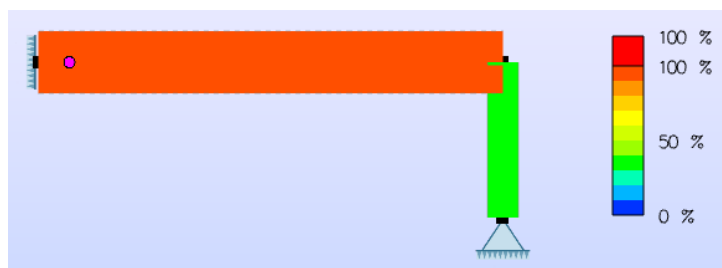
Avgrensningene er nå innført i UM [7, s. 70] og i versjon **3.1** i form av at brukeren varsles dersom det forsøkes å utføre kontroll på konstruksjoner som ikke støttes. Figur 3.63 viser et eksempel.



Figur 3.63: Versjon **3.1**: Advarsel om at dimensjoneringskontrollen ikke støtter den gitte konstruksjonen.

## Component control

*Component control* tar for seg dimensjonering av *members* mht. 2. ordens teori (knekning) [4, s. 351-355] i tegneplanet til konstruksjonene, men ikke ut av planet. Denne kontrollen har på merkelig vis blitt deaktivert i forrige versjon av programmet, uten en grunngivning. Man antar at fjerningen var en menneskelig feil siden det ikke er mulig å oppdrive dokumentasjon for endringen. Kontrollen har blitt reaktivert i versjon **3.1**.

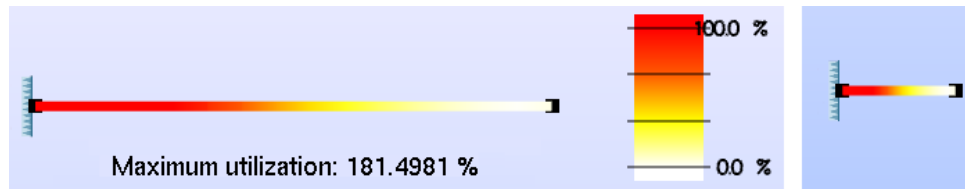


Figur 3.64: Versjon **3.1**: *Component control* er gjeninnført (med ny visualisering).

## Visualisering av resultater

Utnyttelsen av kapasiteten til elementene i konstruksjoner visualiseres i ett diagram for hver dimensjoneringskontroll. Visualiseringen hadde flere svakheter i versjon **3.0**, som er utbedret i versjon **3.1**.





Figur 3.65: Versjon 3.0: Visualiseringen av utnyttelsesgrad var misvisende. Det var f.eks. ikke mulig å skille 90 % fra 100+ %. Konstruksjonen består av 15 elementer.

### Fargeverdi for element

I versjon 3.0 ble fargene interpolert mellom nodene i elementer. Dette gjorde at fargene skiftet nyanser veldig gradvis. Det kunne tilsynelatende se ut som resultatene som presenteres i figur 3.65 var fra en eksakt analytisk løsning (ikke elementmetoden). Interpoleringen av farger gjorde at det ble vanskelig å skille mellom ulike grader av utnyttelse. Figur 3.65 viser at det ikke var helt klart hvor konstruksjonen passerte 100 % utnyttelse. Den røde fargen skulle indikere at det var fare på ferde. Siden den var såpass tydelig, og ble trukket ut over et stort område virket det som om tilstanden var mer kritisk for konstruksjonen enn det som gjaldt.

Algoritmen for å gi farger til elementene er endret. Det gis nå en konstant fargeverdi for hvert element, basert på den gjennomsnittlige utnyttelsen i de to nodene deres. Dog er det slik at om den ene noden har en utnyttelse på 100+ % blir fargen i elementet alltid rød. Ellers gis det en fargeverdi som vist i figur 3.66, med ny farge for hver tiende prosent mellom 0 og 100.

### Fargeskalaen

Fargeskalaen så uryddig ut. Tekst og selve skalaen overlappet. Dette ble rettet i versjon 3.1 og i tillegg ble det lagt til tekst som viser 50 % utnyttelse og maksimal utnyttelse.

### Tallformater for utnyttelsesgrad

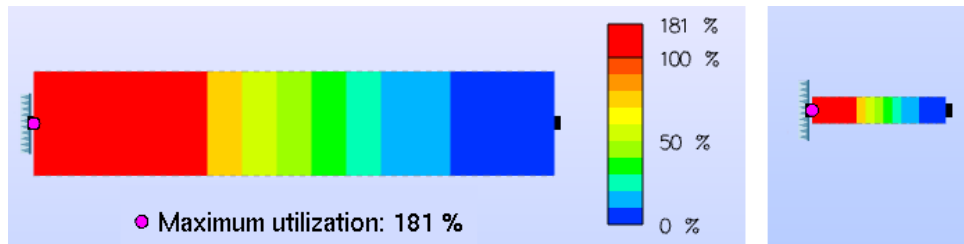
Det ble brukt ulike formater for å vise utnyttelsen: i teksten nede i modellvinduet (prosentvis, med fire desimaler), i fargeskalaen (prosentvis, med én desimal) og i dialogboksen med elementresultater (På desimalform: 0.0 - 1.0). Tallformatet for utnyttelsen er nå den samme både i bunnen av modellvinduet, i fargeskalaen og i dialogen med elementresultater.

### Tegning av elementer

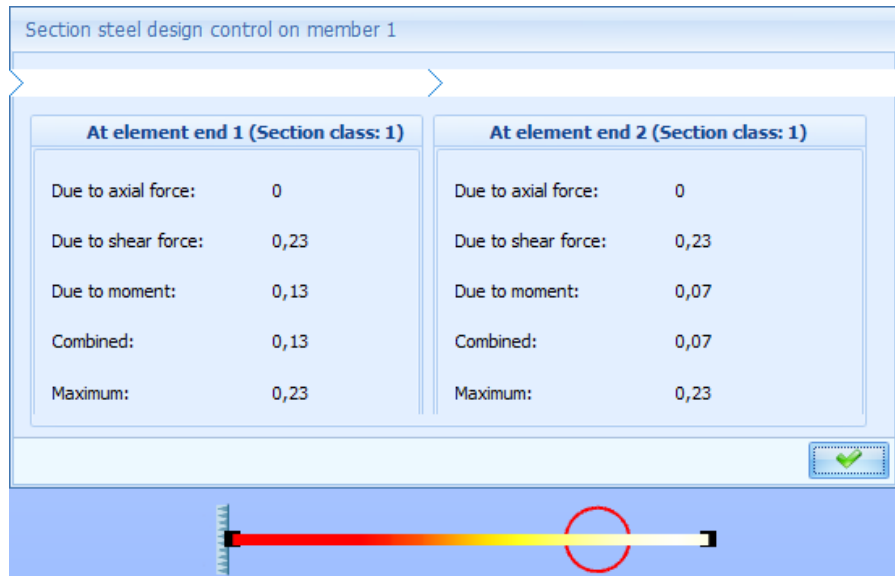
Visualiseringen ble tegnet veldig smalt over *members* og ble utydelig. Bildet t.h. i figur 3.65 illustrerer dette. Visualiseringen tegnes i versjon 3.1 over hele tverrsnittshøyden av elementet. Det er også mulig å markere elementet lettere i hht. vedlegg A.3.5.

### Visning av valgt element

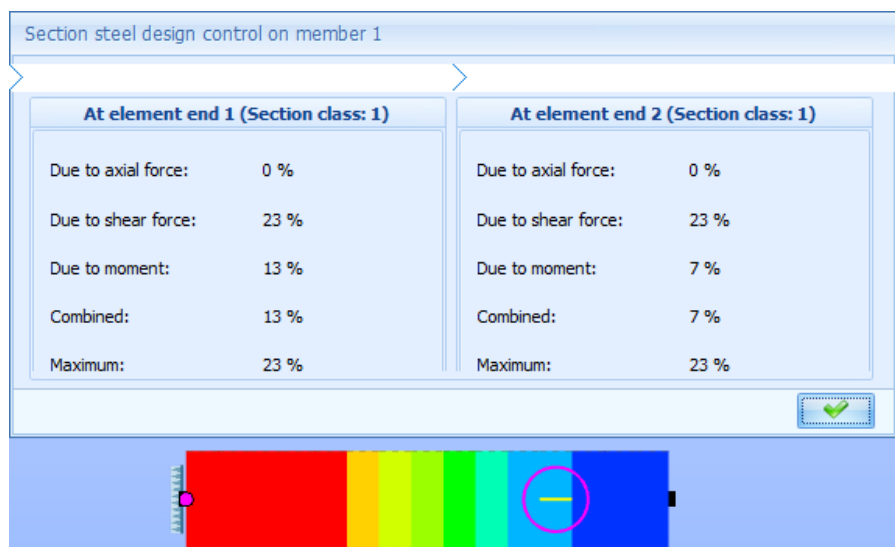
Elementene ble ikke vist som markert fordi diagrammet ble tegnet over markeringen, noe som vises i figur 3.67. Figur 3.68 viser at det valgte elementet er tydeligere markert i den nye programversjonen. I versjon 3.0 var det mulig for brukeren å endre fargeskalaen som skulle benyttes. Den nye fargeskalaen som er innført i versjon 3.1 er konvensjonell med variasjon fra kalde til varme fargetoner, derfor er det ikke behov for å endre den. Til sammen gjør modifikasjonene at resultatene er enklere å tolke, og mer åpenbart basert på elementanalyse.



Figur 3.66: Versjon 3.1: Visualiseringsalgoritmen er endret og presentasjonen er mer informativ. Konstruksjonen består av 15 elementer.



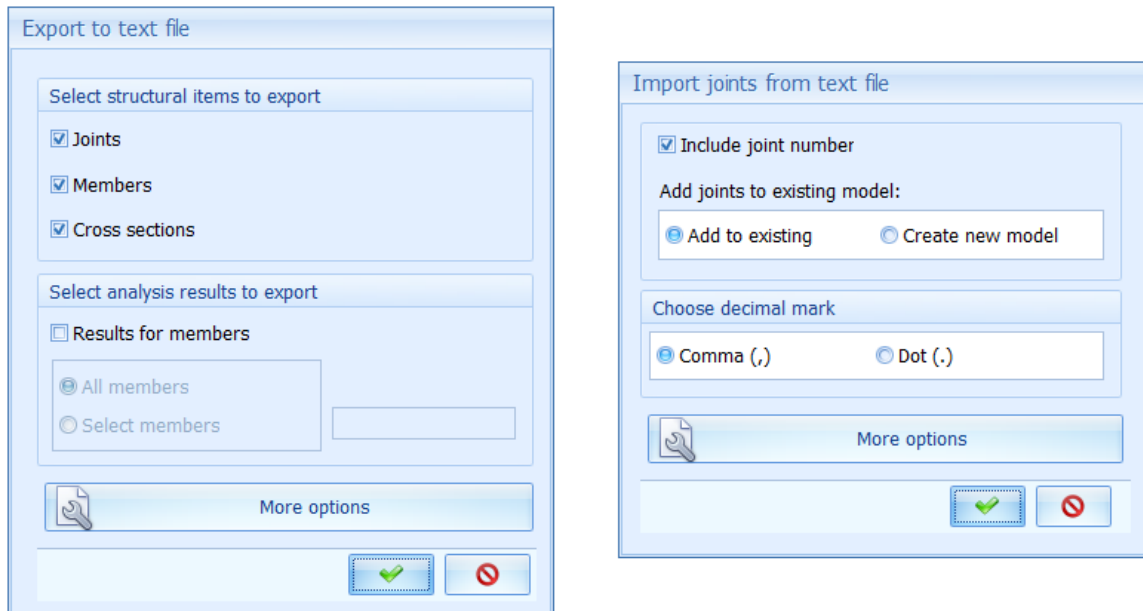
Figur 3.67: Versjon 3.0: Dialog med elementresultater. Utnyttelsen var vist på desimalform. Legg merke til at det er vanskelig å se akkurat hvilket element som er valgt.



Figur 3.68: Versjon 3.1: Dialog med elementresultater. Utnyttelsen vises på prosentform. Det valgte elementet er synlig.

## 3.8 Eksport og import

Eksport og import er nye verktøy i versjon 3.1. Verktøyene behandler tekstfiler i ASCII-format<sup>11</sup> for å tabulere verdier som kan åpnes i andre programmer. Eksport kan også benyttes som en enkel rapportgenerator, siden den raskt kan liste opp detaljer om modellen.



Figur 3.69: Eksport- og importdialogene (noen valg er skjult i figuren).

### Eksport

Eksport skriver ut informasjon om den nåværende aktive konstruksjonsmodellen til et tekstdokument. Funksjonaliteten åpner for å eksportere deler av modellen til andre programmer.

Ytterligere detaljer om eksport og import kan finnes i vedlegg A.8 Eksport og import.

Følgende informasjonskategorier kan eksporteres:

- *Joint*-koordinater
- *Member*-informasjon
- Tverrsnittsdata
- Tverrsnittskrefter for valgte *members*

Informasjonen som beskriver *joints*, *members*, tverrsnitt og krefter deles inn i kolonner slik at det er lett å lese tekstdokumentet. Kolonnebredden er tilpasset til hver enkelt verdi. Feks. vises tallverdier på et spesielt kompakt format (det meste kompakte av desimal- og standardform), med inntil fem signifikante sifre. Tekststrenger kan maksimalt inneholde 20 tegn før de overskrider kolonnebredden.

<sup>11</sup> ASCII er et standard tegnssett som støttes av de fleste grafiske operativsystemer, bla. Windows-PC, Mac og grafiske Linux-distribusjoner.

Hver kolonne blir adskilt av et forhåndsdefinert eller brukerdefinert skilletegn. Som standard er skilletegnet satt til å være et mellomrom.

Et eksportdokument får en overskrift på første linje som gir informasjon om bla. **fap2D**-versjonen som er brukt. Hver informasjonskategori har en tittel og kolonneoverskrifter, med enheter, slik at det ikke er noen tvil om innholdet. Utskriften inneholder de kategoriene brukeren velger å inkludere, men kategoriene kan også skrives ut i separate dokumenter om ønskelig.

Formatet på eksport av *joint*-koordinater er koordinert med import-funksjonaliteten slik at eksport/import-kombinasjonen kan benyttes som en primitiv form for lagring av *joints*. Dette står det mer om i neste avsnitt.

Figur A.144 illustrerer en eksportert tekstfil som inneholder alle kategorier av informasjon.

```
fap2D version 3.5.2 - DE

Joint coordinates

Joint  X-coord. (mm)  Z-coord. (mm)
1      ; 0           ; 0
2      ; 3000        ; 0

Member info.

Member  Type          Geometry  Radius/height (mm)  Joints  Material  Cross sect.
1      ; StraightBeam ; Straight ; 0           ; 1, 2    ; Steel   ; IPE200, Rec 100x100

Cross section info.

Name      Type          A (mm^2)  I (mm^4)  ht (mm)  hb (mm)
IPE 200   ; Predefined ; 2850     ; 1,94E+07 ; 100     ; 100
Rec 100x100 ; Parametric ; 10000    ; 8,3333E+06 ; 50     ; 50

Results for members (section forces).

Results for member 1

Member type:      StraightBeam
Load combination: Default load combination
End joints:       1, 2

s (mm) =         N (kN)          V (kN)          M (kNm)
-5,5565E-15 ; 0           ; -4,2895       ; 5,7925
300          ; 0           ; -3,642        ; 4,5057
600          ; 0           ; -3,0408       ; 3,4131
900          ; 0           ; -2,4858       ; 2,5008
1200         ; 0           ; -1,9771       ; 1,7551
1500         ; 0           ; -1,5146       ; 1,162
1800         ; 0           ; -1,0984       ; 0,70759
2100         ; 0           ; -0,7284       ; 0,37807
2400         ; 0           ; -0,40467      ; 0,15955
2700         ; 0           ; -0,12718      ; 0,038154
3000         ; 0           ; -0,12718      ; 0
```

Figur 3.70: Eksempel på eksportert tekstfil.

## Import

Import-funksjonaliteten i versjon **3.1** kan brukes til å importere *joints* som er gitt ved x- og z-koordinater. Brukeren velger et tekstdokument der hver rad representerer koordinatene og

evt. ID-nummeret til ett *joint*. Dialogen som benyttes for å gjennomføre importering finnes i *application menu*, og vises i bildet t.h. i figur 3.69.

Symbolet som skiller kolonnene i hver rad kan enten finnes automatisk av programmet eller manuelt av brukeren. Den automatiske søkeprosedyren er primitiv ettersom den leter etter skilletegnet i hver eneste rad, men robust siden den kan oppdage de fleste typer skilletegn. Dette er nødvendig fordi programmet ikke alltid oppdager hvilke rader i dokumentet som beskriver *joints*. Den kan også oppdage mellomrom og tabulator dersom den ikke finner andre tegn.

```
fap2D version 3.1
Joint coordinates

Joint  X-coord. (mm)  Z-coord. (mm)
1      1,615E+05     27500
2      2,775E+05     7500
3      4,01E+05      7000
```

Figur 3.71: Eksempel på importfil.

For at programmet skal klare å tolke radene riktig er det viktig at brukeren har angitt korrekt desimaltegn slik at det ikke kan mistolkes som et skilletegn. Komma (“,”) er valgt som standard desimaltegn. Skilletegnet kan være det samme som desimaltegnet dersom det er omgitt av mellomrom på begge sider.

Tallverdiene som leses inn kan både være på desimalform og standardform<sup>12</sup> siden importprosedyren leser inn tall med den innebygde metoden `double.Parse(...)`, som klarer å tolke tall som er skrevet med eksponentiell notasjon.

## 3.9 Settings

Programmets *settings* (innstillinger) er delt inn i to kategorier, *global* og *local settings*. *Local settings* kan settes individuelt for hver enkelt konstruksjonsmodell, mens de globale gjelder hele programmet på tvers av konstruksjonsmodeller. Brukeren kan endre et fåtall *local* og et større antall *global settings*. De fleste lokale er knyttet til beregningsmodellen og vil derfor ikke lagres når **fap2D** avsluttes.

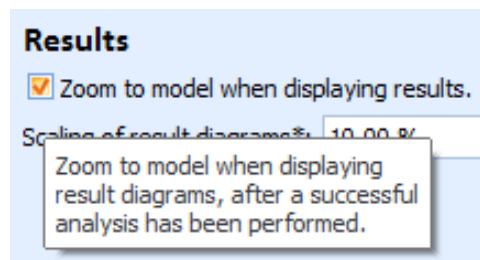
I dette avsnittet presenteres arbeidet som er gjort for å fjerne ubrukte og unødvendige *settings*, og for å legge til nye *settings*. Dokumentasjon fra tidligere utviklere [6, vedlegg A.3.4] viser at det ble gjort solide forbedringer av *settings*-funksjonaliteten i versjon **3.0**, som nå er ytterligere forsterket. Der den gamle *settings*-dialogen kanskje var mest nyttig for intern bruk under utviklingsprosessen er den forhåpentligvis nå bedre tilpasset brukerne.

Vedlegg A.9 gir mer detaljert informasjon om implementasjonene.

<sup>12</sup>Eks. “1.3E5”. Bokstaven “E” må i denne sammenheng ikke forveksles med eulertallet, “e”.

### 3.9.1 Redigering og fjerning av eksisterende *settings*

Dialogboksen med *settings* har blitt rensket for unødvendige valg, og de gjenværende *settings* har blitt sortert i flere kategorier for å skape bedre oversikt. Hver *setting* har i versjon 3.1 fått mer informasjon vha. *tooltips*, i sammenheng med den systematiske innsatsen for å forbedre *tooltips* som er nevnt i kapittel 3.11.1. Flere punkter i dialogboksen for *settings* var rett og slett ikke i bruk og burde vært fjernet tidligere. Dessuten ble også flere unyttige *settings* som lå skjult i kildekoden fjernet.



Figur 3.72: Eksempel på *tooltip* i *settings*.

#### *Steel design*

Visualiseringen av utnyttelse i hht. ståldimensjonering hadde tidligere mange tilgjengelige *settings*. Etter at visualiseringen ble strammet inn i versjon 3.1 måtte disse valgene fjernes.

### 3.9.2 Lagring av *settings*

*Global settings* lagres til mappen *Appdata* som er forbundet med brukerens Windows-konto. *Appdata* er et vanlig sted å lagre brukerinnstillinger for programvare, derfor er det et naturlig å lagre *global settings* for **fap2D**. En fordel med å lagre *settings*-filen i en mappe tilknyttet brukeren er at innstillingene vil følge brukeren selv om det benyttes en annen PC på samme PC-nettverk.

Tidligere ble *global settings* lagret til installasjonsmappen i programfilene. Bakdelen med den praksisen var at man trengte administrasjonsrettigheter for å la programmet skrive til mappen etter hver analyse. Ved institusjoner som NTNU er det ikke vanlig at f.eks. studentbrukere har *admin*-retter, noe som skapte problemer under betatesting.<sup>13</sup>

Den nye mappestien til filen er:

```
C:\Users\\Appdata\Roaming\NTNU\fap2D\global.fap
```

### 3.9.3 Nye innstillinger

#### *Units - enheter*

Flere nye enheter er lagt til etter ønske og behov. Det er nyttig å kunne bruke ulike sett med

<sup>13</sup>Betatestingen beskrives i vedlegg B Betatesting av fap2D 3.0.0.

dimensjoner for å modellere problemer av ulik størrelsesorden, uten å måtte gjøre mange omregninger. Av erfaring vet man at det lett oppstår regnefeil når man må utføre mange manuelle omregninger. Tabell 3.2 viser nye enheter som er lagt til.

Tabell 3.2: Enheter som nå kan endres i *settings*.

Størrelse	Standard	Lagt til
Frekvens	<b>Hz</b>	rad/s
Rotasjonsstivhet	<b>kNm/deg</b>	Nmm/deg, kNm/rad
Aksialstivhet	<b>kN/m</b>	kN/mm, N/mm
Kraft/lengde	<b>kN/m</b>	N/mm, N/m
Spenninger	<b>N/mm<sup>2</sup></b>	MPa, $N/m^2$
Demping	<b>kNs/m</b>	Ns/mm, Ns/m
Areal	<b>mm<sup>2</sup></b>	$m^2$
2. arealmoment	<b>mm<sup>4</sup></b>	$m^4$

### ***Zoom to model when displaying results***

Brukeren kan velge om det automatisk skal *zoomes* slik at hele modellen vises når resultatene for en gjennomført analyse skal vises.

### ***Snap og selection settings***

Modifikasjonene av *snap* og *selection*, beskrevet i avsnitt 3.3 Markering av *structural items* har et par tilhørende *settings*. Brukere får mulighet til å definere hvor presis en markering av *structural items* skal være før de blir valgt. De kan også bestemme størrelsen på området rundt en *arch member* som kan brukes til å markere den. Det påpekes at det ikke er nødvendig å endre disse innstillingene med mindre det er vanskelig å markere visse *items*.

### **Advarsler**

Det er innført advarsel om numerisk ustabilitet pga. korte elementer og advarsel om tvilsom *arch member*-geometri (spisse parabelbuer) som guider brukeren til å konstruere nogenlunde gyldige modeller. Advarslene kan oppleves som distraherende fordi de kan dukke opp ofte. Derfor er det mulig å skjule dialogboksene i *settings*. Advarslene er aktive som standard.

### **Gravitasjonsakselerasjon, g**

Verdien for gravitasjonsakselerasjon, **g**, kan endres. Standardverdien er  $9,81 m/s^2$ . Verdien brukes i beregningen av egenlast og kan variere i forhold til at den varierer fysisk sett på overflaten av jordkloden.

### ***Include own weight on default***

Dersom innstillingen er aktivert vil egenvekt legges til i *default load combination* når det ikke er lagt til noen eksterne laster i modellen. Om innstillingen ikke er aktivert vil brukeren få valget mellom å inkludere eller ekskludere egenvekt hver gang det ikke er lagt til andre laster i modellen. Standardinnstillingen er inaktiv.

### ***Merge overlapping joints on default***

Man kan velge om *joints* som overlapper med hverandre automatisk skal slås sammen. Standardinnstillingen er at de ikke skal det.

### 3.10 Validering av input

Dette avsnittet tar for seg forsøket som er gjort på å innføre mer systematisk validering av brukerinput i **fap2D**. Valideringen av spesifikke problemer, slik som *arch members* er omtalt andre steder i kapitlet.

Det er to hovedgrunner til at validering har blitt prioritert: den første er at det reduserer faren for programstans (pga. *unhandled exceptions*), den andre er at utviklere unngår å bruke mye tid på å kontrollere input-verdier hver gang de skal brukes. Det systematiske arbeidet som er gjort for å øke valideringen omfatter kontroll av tegnene som kan brukes i tekstfelter og kontroll av gyldigheten til input-verdier etter at de er blitt skrevet inn.

For at numeriske inputverdier skal angis korrekt er Mask-egenskapene til tekstfeltene (*TextEdits* fra DX-rammeverket) blitt benyttet. Masks gjør det mulig å kontrollere hvilke og hvilket antall tegn som benyttes. De sikrer også bruk av riktig desimaltegn. Det er også laget og benyttet metoder som sørger for at tallverdiene som angis er gyldige for sitt bruksområde. Eksempelvis at visse verdier ikke kan settes lik "0".

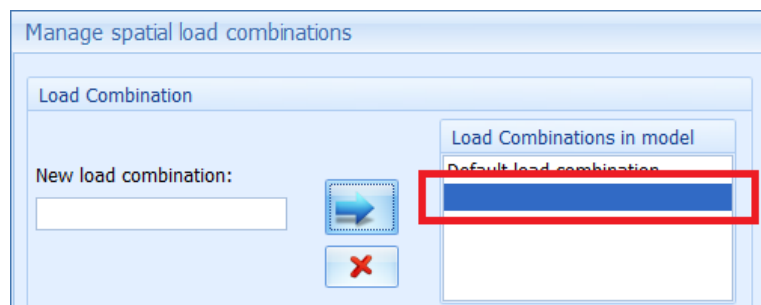
Figuren under viser eksempel på validering av numerisk input vha. en *parser* (tolker-metode) i **fap2D**:

```
private void txtN_per_Validating(object sender, System.ComponentModel.CancelEventArgs e)
{
    int number;

    if (int.TryParse(txtN_per.Text, out number))
        if (number < 1)
            txtN_per.EditValue = 1;
}
```

Figur 3.73: Koden viser bruk av `int.TryParse(...)` som er en metode som sjekker om input kan representeres som et heltall (*integer*). Hvis input ikke kan det, vil det ikke gjennomføres noen endringer. Programmet beholder dermed kun gyldige verdier. Tilsvarende metoder er implementert for flyttall.

Tekstfelter for navn på f.eks. *load combinations* og *load cases*, og annen tekstlig input valideres slik at de ikke kan settes til en tom streng (" "). Dette er implementert for å unngå at felter ikke blir opprettet i lagringsfilene til modellene. Tomme strenger skapte nemlig et problem ved innlasting av modellfiler som er forklart i avsnitt 3.12 Filbehandling.



Figur 3.74: Versjon 3.0: En *load combination* har blitt lagret med en tom streng som navn.



## 3.11 Informasjon til brukeren

Dette avsnittet viser forbedringer som er gjort for å gi brukeren bedre tilleggsinformasjon slik at det er enklere å gjøre valg i programmet. Fokusområdet har vært *tooltips*, der det har blitt brukt tallrike timeverk.

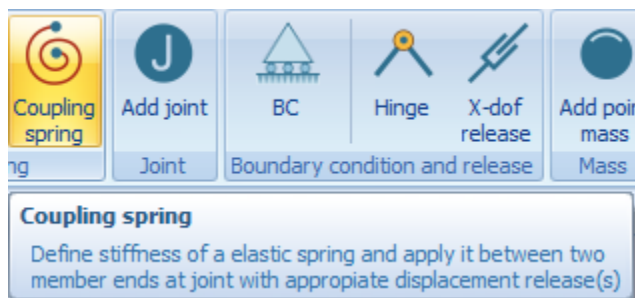
### 3.11.1 Tooltips

*Tooltips* er en funksjon som tilfører mer informasjon til *controls* i GUI. I bunn og grunn er et *tooltip* bare en liten hjelpetekst som dukker opp når man holder musepekeren over en *control*. Det er lagt stor vekt på å forbedre disse, slik at ferske brukere raskt kan lære seg å bruke ulike deler av programmet.

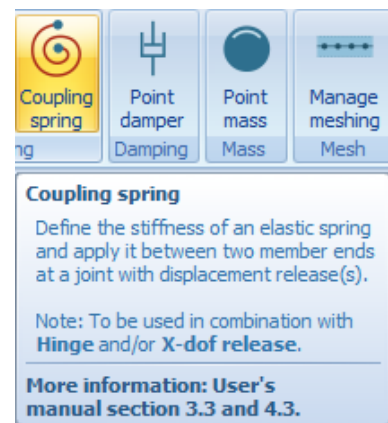
Ved å innføre gode *tooltips* blir man mindre avhengig av å lese i UM, og behovet for *tutorial* blir mindre. Dersom brukeren har behov for informasjon utover den som vises i *tooltips* er det i mange tilfeller referert til relevant kapittel i UM.

De fleste *tooltips* har blitt revidert for å gi økt lesbarhet og mer presis informasjon. Det er også meningen at *tooltips*-ene skal følge samme mønster og ordlyd for helhetens skyld. *Controls* som kan betjenes vha. hurtigtast nevner nå hurtigtasten i sitt *tooltip*.

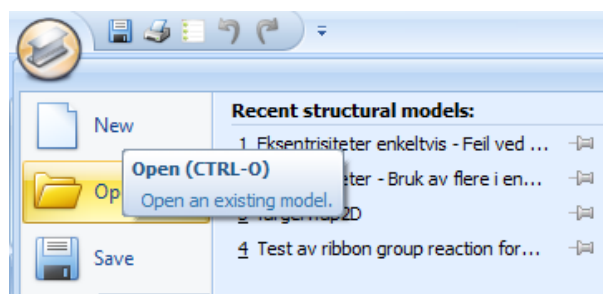
Dessuten er det lagt til en del nye *tooltips* for å komplettere programmet. Mer informasjon finnes i vedlegg A.11.1. En fullstendig oversikt over revisjonen er gitt i vedlegg B.2.6.



Figur 3.75: Før endring av *super tooltip*.



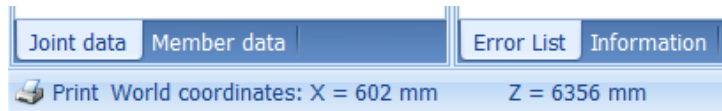
Figur 3.76: Etter endring av *super tooltip*.



Figur 3.77: Etter endring: Eksempel på nytt *tooltip*, med hurtigtast.

### 3.11.2 Musepeker-kordinater i *status bar*

Koordinatene til musepekeren vises nå i *status bar*, nederst i GUI. Det er lagt til et felt som skriver ut pekerens nåværende posisjon i modellvinduet, gitt i *world coordinates*, dvs. pekerens posisjon i modellens koordinatsystem. Koordinatene avrundes til nærmeste heltall, det er presist nok til at brukeren ser hvor man befinner seg i modellen.



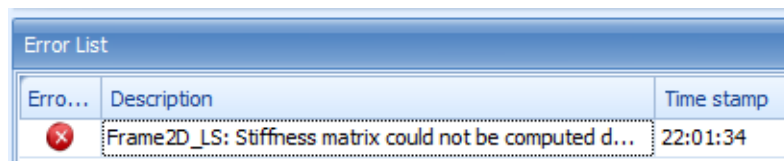
Figur 3.78: Musepeker-kordinater i *status bar*, gitt i *world coordinates*.

Ved å holde et øye med *status bar* er det lettere å plassere *joints* og *members* korrekt. Det fører igjen til færre tastetrykk og bedre flyt. Man kan si at denne endringen derfor kompletterer *snap to grid*-funksjonen og gjør brukeren mindre avhengig av å gjøre justeringer vha. *joint data*-tabellen.

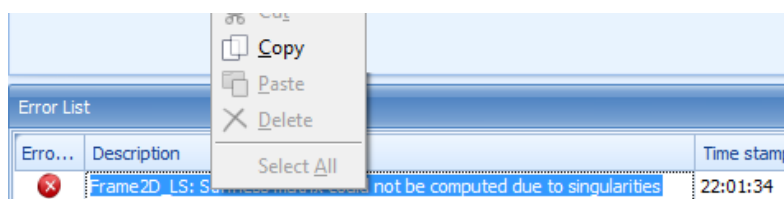
### 3.11.3 Formatering av *error* og *information lists*

Formateringen av informasjon som vises i *information list* har blitt forandret for å gjøre den mer oversiktlig og enhetlig. Teksten er gjort grammatisk konsekvent, data vises etter kolon istedenfor i parentes, og informasjon om forskjellige analyser skilles med en blank linje. Den siste linjen som vises for hver analyse er et tidsstempel. For å se detaljerte beskrivelser av endringen bør man ta en kikk i vedlegg A.11.2.

*Error list*, som viser feilmeldinger, er blitt mer informativ og enklere å bruke. Hver melding markeres med et tidsstempel for å skille feilene fra hverandre. Dette er den viktigste endringen, fordi tidsstempelet gjør at man kan se hvilke feil som oppstår samtidig, og derfor har sammenheng. Kolonnene kan justeres for å vise meldinger som vanligvis ville blitt for korte for listen. I tillegg kan feilmeldingene nå markeres som tekst slik at brukeren kan kopiere dem. Figur 3.79 og 3.80 viser modifikasjonen.



Figur 3.79: Versjon 3.0: Teksten i *error list*: kan ikke kopieres.



Figur 3.80: Versjon 3.1: Teksten i *error list* kan nå kopieres.

### 3.11.4 Tusen-skilletegn og prosentvisning i tallverdier

For at det skal være lettere å lese store tallverdier er det lagt til et mellomrom mellom hver tusen i tekstfelter som viser input- og output-verdier. Figur 3.81 illustrerer poenget. Se vedlegg A.11.6 for detaljer.

Endringen er forøvrig sterk forbundet med vedlegg A.10.1.



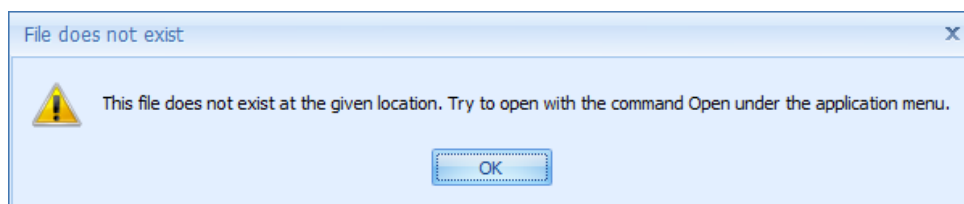
Figur 3.81: Før og etter at tusen-skillet ble lagt til.

En annen endring som er gjort fordi den er mer intuitiv og lettleselig er å benytte prosentformat for å vise utnyttelsen ved ståldimensjonering. Prosentformatet ble innført i resultatvisningen i versjon **3.0**, men er nå også blitt innført i *steel design usage*-dialogen. Figur 3.68 viser dialogen etter endring.

## 3.12 Filbehandling

### Fjerning av slettede modellfiler

Listen over *recent structural models* i *application menu* skal vise de siste filene som har blitt brukt i programmet. Dersom en av filene som ligger i listen ikke eksisterer på sin opprinnelige filplassering lenger, bør den fjernes fra selve listen. Det skjedde ikke i versjon **3.0**, men er rettet i versjon **3.1**. Dessverre oppdaget man at implementasjonen ikke fungerer i alle tilfeller, som beskrevet i vedlegg A.12.1.



Figur 3.82: Varsel om at en fil er flyttet eller slettet fra sin opprinnelige plassering.

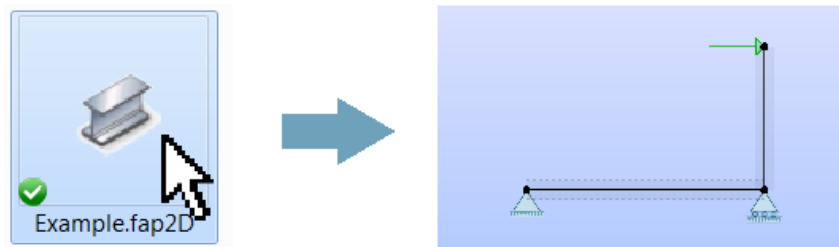
### Åpning av modellfiler

I versjon **3.0** ble **fap2D** åpnet på samme måte uansett om programmet ble startet vha. programikonet, eller vha. å trykke på en assosiert modellfil (".fap2D"-fil). For sistnevnte tilfelle ble altså velkomsts skjermen vist selv om den ikke var nødvendig, fordi man alt hadde valgt hvilken fil som skulle åpnes. I den nye versjonen vil man hoppe rett inn i den ønskede modellen når man trykker på en eksisterende modellfil i f.eks. Windows Utforsker.<sup>14</sup> Se vedlegg A.12.3 for detaljer.

<sup>14</sup>Windows Utforsker er det grafiske grensesnittet som brukes til å bla i filsystemet på en Windows-PC.



Figur 3.83: Før endring: Åpning av eksisterende modell fungerer ikke.

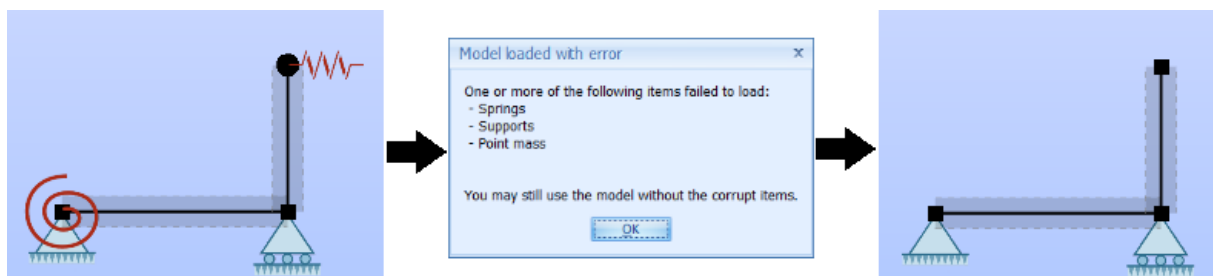


Figur 3.84: Etter endring: Åpning av eksisterende modell fungerer som ønsket.

### Lagring og lastning av filer

Lagring og innlasting av filer er blitt forbedret. Programmet vil nå takle feil og mangler i modellfilene ved å lagre og laste alle gyldige *structural items*, og gi bruker varsel om eventuelle feil. Alle *joints* og *members* må være gyldige for at modellen skal lagres og lastes, men alle andre *items* kan ignoreres hvis de inneholder feil. Modellen kan da fremdeles benyttes, og bruker vil bli gjort klar over manglene. Figur 3.85 illustrerer dette.

Denne løsningen var en middelvei mellom metoder som kun godtar 100% korrekte modeller, og metoder som går i hver detalj og forsøker å fikse eller melde fra om dem. Detaljert diskusjon om problemstillingen og løsningen er gitt i vedlegg A.12.2 Forbedret lagring og lastning av filer.



Figur 3.85: Etter endring: Varsel om delvis feil ved lastning av modellfil.

# Kapittel 4

## Tilstandsrapport

Dette kapitlet vurderer den nåværende og fremtidige tilstanden til **fap2d**. Først presenteres drøftinger av prosjektet og programmet **fap2D** og til slutt presenteres dokumentasjon som kan komme fremtidige utviklere og brukere til nytte.

På det nåværende tidspunktet antas det at prosjektet **fap2D** er på veg over fra en utviklingsfase til en bruksfase. Likevel er det fortsatt rom for videreutvikling dersom noen er villige til å fortsette arbeidet. Uavhengig av om det skjer er det viktig å videreføre kunnskapen om programmet til fremtidige brukere og utviklere.

### 4.1 Betatesting av fap2D 3.0.0 Beta

Hensikten med betatesting er å se om brukere i programmets målgruppe er fornøyde med hovedfunksjonene i programmet. Samtidig er det viktig for utviklerne å få tilbakemelding om feil og mangler. Det er vanlig å bruke et feilrapporteringsystem for å få tilbakemeldinger fra testgruppen.

Det har blitt gjennomført betatesting av **fap2D 3.0** i forbindelse med øvinger i faget TKT4180 Konstruksjonsmekanikk - Beregningsmetoder (KMEK).<sup>1</sup> Veileder har holdt demonstrasjon av programmet for studentene, og vi har stilt opp som *support/støtte* for å svare på spørsmål og henvendelser. Et feilrapporteringskjema ble delt med studentene og en evaluering av versjon **3.0** ble gjennomført.

Om lag tjue feilrapporter og henvendelser, og eksakt 17 evalueringsskjemaer ble registrert. Man håpet på flere skjemaer, men må ta selvkritikk for at skjemaet ble lagt ut etter forelesningslutt i faget. De sytten evakueringen ga likevel et visst grunnlag for å trekke slutninger fra svarene i dem.

Evalueringene ble benyttet aktivt for å avgjøre hvilke endringer og nye implementasjoner i programmet som skulle gjennomføres.

Før betatestingen begynte var det knyttet skepsis til om studentene ville bli tilfredse med versjon **3.0**. Etter at alle som evaluerte den sa at de benyttet **fap2D** til fordel for alternativet **Focus Konstruksjon 2D** [12], ble versjonen vurdert som høyst brukbar. Likevel må man ta høyde for at de som svarte er de som er mest positivt innstilt til **fap2D**.

Betatestingen tolkes som en suksess og neste versjon, **3.1**, slippes som en ordinær utgivelse, ikke en betaversjon. Vedlegg B Betatesting av fap2D 3.0.0 viser hele grunnlaget for denne vurderingen.

---

<sup>1</sup>Pedersen oppfordret til betatesting i avsnitt 6.8 i [9].

## 4.2 Distribusjon, lisensiering og installasjon

Utviklerteamet har diskutert hvordan **fap2D** bør lisensieres og distribueres. Per nå eksisterer det ikke noen ordentlig lisenskontroll og distribusjonen kan ikke kontrolleres.

Det er ikke gjort forsøk på å utvikle en lisensieringsmetode, men man er enig med Pedersen [9, kap. 6.4] i at distribusjon via en NTNU-server ville vært gunstig. Programmet vil bli distribuert for bruk internt av studenter og ansatte ved KT, men KT tar ikke noe ansvar for bruk av programmet, som presisert i UM.

Det er også blitt diskutert hvordan installasjonsprosedyren skal fungere. Slik det er lagt opp nå må man installere tilleggsprogramvare for at **fap2D** skal kunne brukes.<sup>2</sup> Det hadde vært praktisk å få lagt til alle tilleggsprogrammene i den samme installasjonsfilen, men betatestingen viser at de fleste brukerne ikke har hatt problemer med installasjon, så sant de har fulgt retningslinjene som har blitt gitt. Nytteverdien av en samlet installasjonsprosedyre vil derfor koste mer tid, og sannsynligvis penger, enn det er verdt.

I samråd med veileder har det blitt laget tre ulike utgaver av versjon **3.1**, som sikrer at programmet kan brukes i overskuelig framtid. Én utgave har lisens fram tid september 2015, slik at den kan brukes på kort sikt av studenter ved KT. Den andre utgaven har fri lisens. Den siste utgaven har fri lisens, og skriver ikke ut loggfil, slik at den kan brukes på datasaler på NTNU.<sup>3</sup>

## 4.3 “3D-effekter” i fap2D

**fap2D** er laget som et todimensjonalt program. Likevel er det noen “3D”-effekter som oppstår. Undervegs i utviklingen har det vært situasjoner der man har måttet velge om “3D”-effektene fortsatt skal oppstå. I hvilken grad implementasjonen av disse effektene har vært bevisst er usikkert. Nedenfor redegjøres det for om de bør beholdes, og i så fall hvordan.

### 4.3.1 Overlappende *members*

Oppførselen til overlappende *members* vises i eksempel 1 og 2, vist i hhv. figur 4.1 og 4.2.

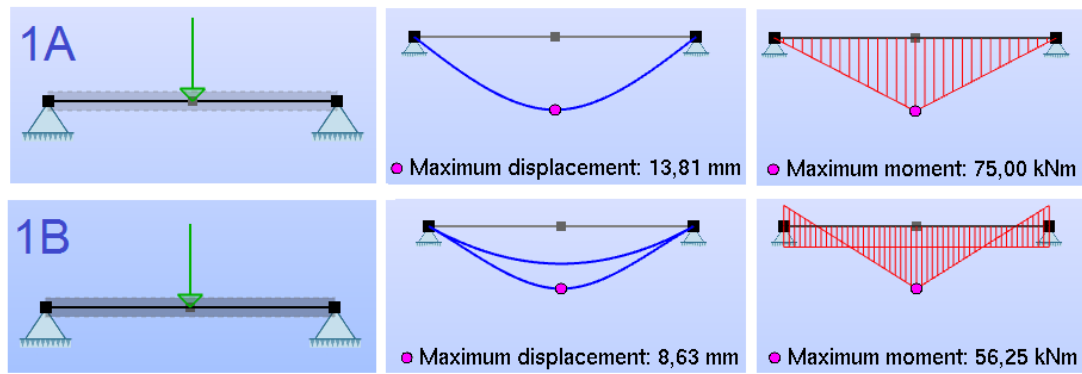
Eksempel 1 viser en sammenligning av fritt opplagte *members* der 1A har én *member*, og 1B har to *members*. Én *member* i hvert tilfelle har punktlast midt på spennet. Eksempel 2 viser en sammenligning av fritt opplagte *members* med en utkrager. 2A og 2B har hhv. én og to *members* i spennet. Både 2A og 2B har en punktlast ytterst på utkrageren. I begge eksemplene kan man observere at de overlappende *members* har mørkere *member outline* slik at de skiller seg ut.

I tillegg til at det er mulig å lage helt overlappende *members* er det, som vist i figur 4.3, også mulig å lage modeller med delvis overlappende *members*.

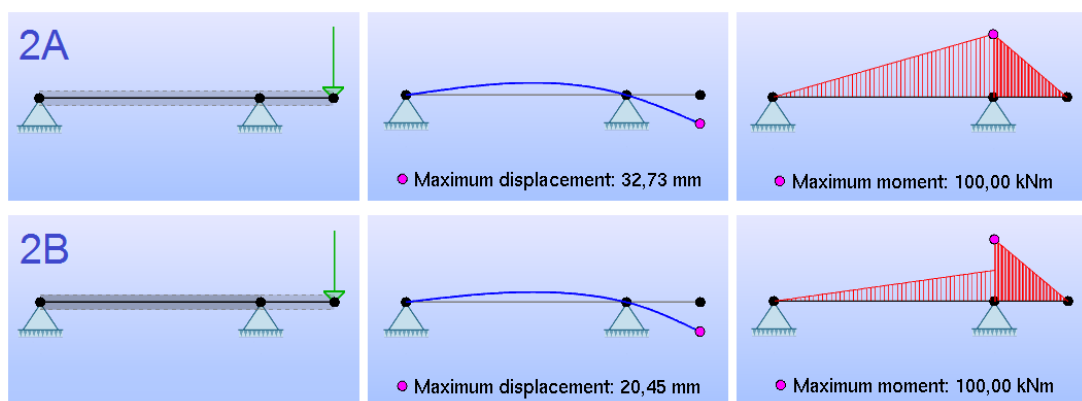
---

<sup>2</sup>Se vedlegg F.1 Systemkrav for fap2D.

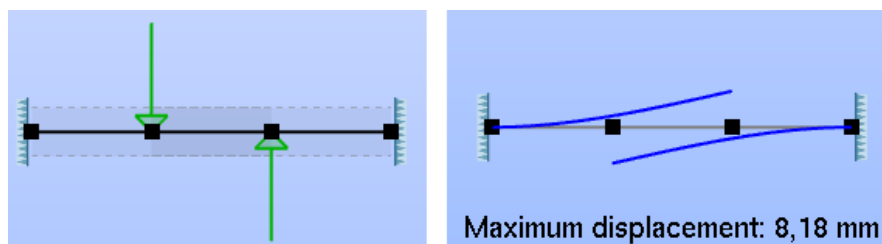
<sup>3</sup>Betatestingen avslørte et problem med loggfilen, som er nevnt i vedlegg B Betatesting av fap2D 3.0.0.



Figur 4.1: Eksempel 1 - 1A: Kun en *member* - 1B: Overlappende *members*.



Figur 4.2: Eksempel 2 - 2A: En enkel *member* i spennet - 2B: Overlappende *members* i spennet.



Figur 4.3: Eksempel: Delvis overlappende *members*.

Resultatene viser at overlapp har effekt ved at det stiver opp modellen. I eksempel 1 tar *member*-en uten last opp konstant bøyemoment som tilsvarer bøyemomentet i endepunktene til *member*-en med last. Diagrammene i eksempel 2 viser samme oppførsel, men her fordeles bøyemomentet fra utkrageren likt til begge de overlappende *members*.

Oppførselen til overlappende *members* virker å være konsekvent. De fungerer som vanlige, separate *members*, lagt i ulike plan og knyttet sammen med stive knutepunkt. Problemet er at det kan være vanskelig å se at *members* overlapper, og vanskelig å kontrollere hvem av dem som påsettes last. Til tross for dette er det ikke vurdert nødvendig å forandre oppførselen siden den gjør **fap2D** mer fleksibelt.

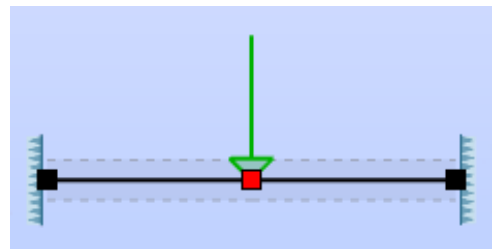
### 4.3.2 Overlappende *joints*

Det er også mulig å modellere overlappende *joints* i programmet. Det har tidligere blitt laget svært nyttig logikk som registrerer overlappende *joints* og endrer fargen deres til rød, slik som i figur 4.4. Dermed er det ingen tvil overfor brukeren om tilstanden deres. Dog skal det presiseres at for å registrere to *joints* som overlapper må de ha nøyaktig lik posisjon, inkludert alle (ca. 16) signifikante sifre. I praksis betyr det at det er lett å lure prosedyren, men brukere vil neppe ha til hensikt å gjøre det, og vil ved vanlig bruk klare å unngå det.

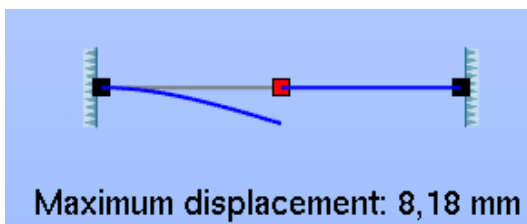
Eksempelet vist i figur 4.5, 4.6 og 4.7 illustrerer en effekt som kan skapes ved å benytte overlappende *joints*, og som kan tenkes å være nyttig.



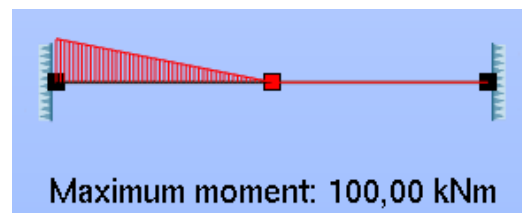
Figur 4.4: To overlappende *joints* t.h i *member*.



Figur 4.5: Eksempel som illustrerer en effekt av overlappende *joints*. Kun en *member* vil få last påsatt.



Figur 4.6: Forskyvning pga. lasten i figur 4.5.



Figur 4.7: Moment pga. lasten i figur 4.5.

Det må presiseres at for å konstruere overlappende *joints* må man først lage dem separat og deretter flytte et av dem til det andres posisjon. Det bør også nevnes at det ikke er mulig å enkelt kontrollere hvilket av de overlappende *joints* som får last påsatt.

Som følge av observasjonene er det vurdert at det ikke er nødvendig å hindre konstruksjon av overlappende *joints*. Det er likevel lagt til en ny innstilling som gjør at bruker kan slå sammen overlappende *joints* når tilfellet oppstår. Denne innstillingen, som er nevnt i vedlegg A.4.10 og A.9.9, gir brukeren mer kontroll over oppførselen.

## 4.4 Videreføring av kunnskap

I prosjekter og organisasjoner er kunnskapsdeling viktig. Deling setter føringer for effektivitet, samordning, og personlig og felles utvikling.

**fap2D** som et prosjekt står i en noe ugunstig posisjon på dette området. Det er bare veilederen i prosjektet som har vært med i hele fartstiden til prosjektet. Utviklerne som



arbeider med GUI skiftes ut hvert år. Derfor er det essensielt at kunnskap overføres på en god måte som uavhengig av om tidligere utviklere er tilgjengelige.

Det finnes ulike virkemidler for å videreføre kunnskapen, men for **fap2D** er det viktigste at det finnes dokumentasjon av endringer som blir gjort i kildekoden til programmet. Dokumentasjonen bør både være i form av overordnet redegjørelse for valg som gjøres, samt begrunnelser for spesifikke implementasjoner i kildekoden.

Videreføringen av kunnskap i prosjektet **fap2D** kan vurderes som svak, basert på følgende begrunnelse:

- GUI-utviklerteamet skiftes ut hvert år. Selv om det er mulig for å kontakte tidligere utviklere må det tas høyde for begrenset tilgjengelighet og at kunnskap kan være glemt fordi de ikke lenger er direkte involvert.
- Versjonskontrollen for kildekoden skiftes ut hvert år, og det finnes ikke et integrert *bug tracking system*. Det hadde vært fordelaktig å ha tilgang til versjonskontrollen fra dag 1 i prosjektet slik at man kan se et langsiktig bilde av endringene i kildekoden. Et integrert *bug tracking system* ville skapt en tettere kobling mellom feil som blir rapportert og måten de er rettet på i programmet. Per i dag må man holde oversikt over *bugs* separat og forholde seg til flere systemer for å ha oversikt over progresjonen.
- Det meste av skriftlig dokumentasjon i prosjektet videreføres i form av masteroppgaver og det er ikke et passende format. Den bør i høyere grad videreføres på samme måte som enkeltstående, redigerbare dokumenter slik at dokumentasjonen kan oppdateres i stedet for å måtte skrives på nytt. Erstad, Pedersen og Aasmundrud startet gode initiativer med dokumentene “Kodekonvensjoner” og “Oppsett av versjonskontroll”. Det fantes lignende dokumenter tidligere, men de var enten utdaterte eller utilstrekkelige.

Virkemidler for å økt videreføring av kunnskap i fremtiden:

- Videreføring av versjonskontroll, med *bug tracking system*. Dersom det aksepteres at kildekoden kan lagres eksternt fra NTNU foreslår vi at prosjektet tar i bruk utviklingsmiljøet Visual Studio Online, med integrert versjonskontroll og *bug tracking*. Visual Studio Online er en skytjeneste-versjon av Visual Studio. Skytjenesten er gratis for inntil 10 studentbrukere (per april 2014). [15]
- Fortsette å dokumentere og lage oversikter over programmet i samme bane som Erstad, Pedersen og Aasmundrud startet.

## 4.5 Testing og validering

Uttrykkene validering og verifisering brukes ofte når man skal vurdere om noe som testes fungerer korrekt. Det understrekes at det stort sett er foretatt validering av modellering og endringer i denne masteroppgaven. Dvs. at modellering og endringer har blitt vurdert for å se om de følger spesifikasjonene for programmet. Det er fremdeles et stykke unna å kunne si noe om at programmet er verifisert.

### 4.5.1 Analyser

Analysene i **fap2D** er validert i henhold til programspesifikasjonene av personene som har implementert de enkelte analysene. De bør likevel testes mer omfattende, men testing av analyse har blitt nedprioritert i denne masteroppgaven til fordel for retting av modelleringsprosessen. Det er lagt ved noen få eksempler fra dynamisk analyse i vedlegg B.2.6, men disse er nyttigere som *tutorials* for brukere enn som validering av analyser.

Likevel er det gjennomført noen enkle valideringer for de områdene som har blitt endret i konstruksjons- og beregningsmodellen.

Det bør muligens vies en egen, mer teoretisk rettet masteroppgave, eller flere prosjektoppgaver, til arbeidet med validering av analyser. Dersom **fap2D** benyttes i øvingsopplegget i flere fag ved KT vil det bidra til testing og validering.

### 4.5.2 Validering av *arch members*

#### Oppsummering av problemstilling

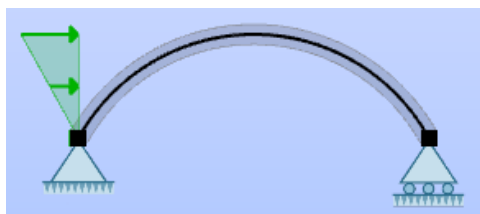
For å validere beregninger problemstillingen som presenteres i avsnitt 3.5.3 er resultatene av analyse i versjon 3.0 og 3.1 blitt sammenlignet. Antagelsen i problemstillingen var kor fortalt at det oppstår upresis lumping av linjelaster i *arch members*, men at fin elementinndeling i praksis fjerner unøyaktigheter. Resultatene var i samsvar med antagelsen for det presenterte eksempelet, og man antar at dette gjelder på generelt grunnlag. Det skal dog bemerkes at det ikke har vært gravd dypt siden problemstillingen ble oppdaget på et sent tidspunkt.

#### Sammenligning av eksempelmodell i versjon 3.0 og 3.1

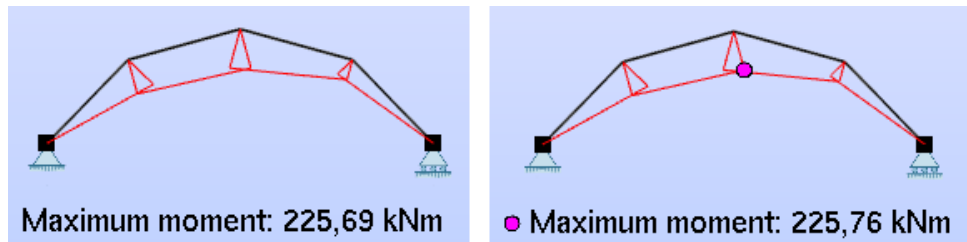
Eksempelmodellen er en fritt opplagt *circular arch member*, med buelengde,  $L = 4261$  mm, radius,  $r = 2000$  mm, materiale = stål (kvalitet S355) og tverrsnitt = IPE 200. Linjelasten som er satt på har intensitet = 300 kN/m øverst i buen, som vist i 4.8.

Først ble *member*-en modellert med 4 elementer, som vist i figur 4.9. Det oppstod som forventet et avvik mellom versjon 3.0 og 3.1, men det var svært lite: det relative avvik mellom maksimalt bøyemoment er mindre enn 0,05% for det aktuelle tilfellet. For 10 elementer var det ingen synlige avvik mellom versjon 3.0 og 3.1 i de fem første signifikante sifrene.

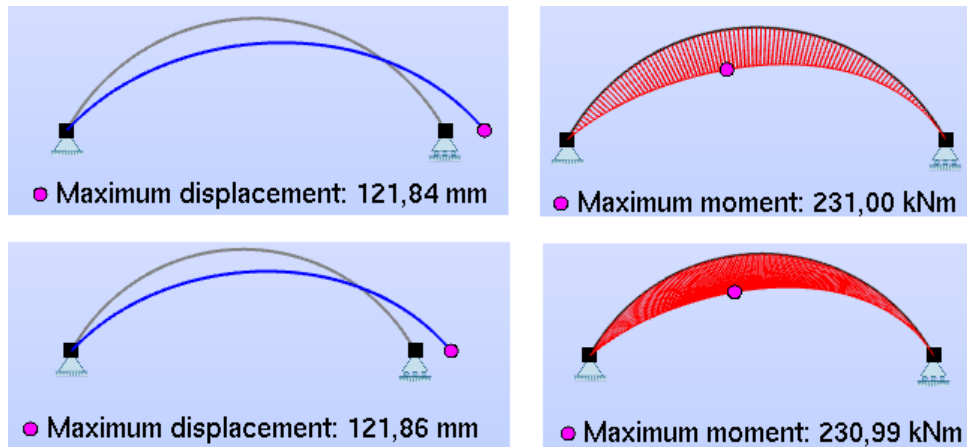
For å vurdere om usikkerheten i beregningene av linjelaster har betydning, ble konvergens for responsen i modellen testet, som vist i figur 4.10. Maksimal forskyvning for *arch member*-en konvergerer med 0,02 mm når antallet elementer økes fra 100 til 200. Dette viser at usikkerheten i praksis ikke spiller noen rolle.



Figur 4.8: Eksempelmodell av *circular arch member* m/linjelast.



Figur 4.9: Avvik mellom momentdiagrammer i hhv. versjon 3.0 (t.v) og versjon 3.1 (t.h).



Figur 4.10: Konvergens av forskyvnings- og momentrespons for *circular arch member* i versjon 3.0/3.1. Hhv. 100 elementer øverst og 200 nederst.

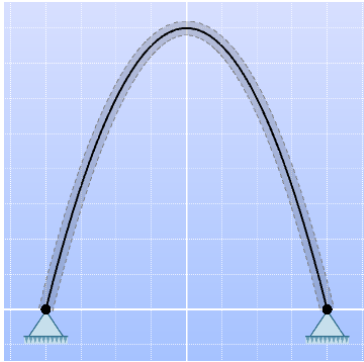
### Ytterligere testing

Det ble som nevnt begrenset tid til valideringen. Derfor bør det undersøkes hvordan *arch members* oppfører seg for ulike typer linjelaster og ved bruk av *parabolic arch members*.

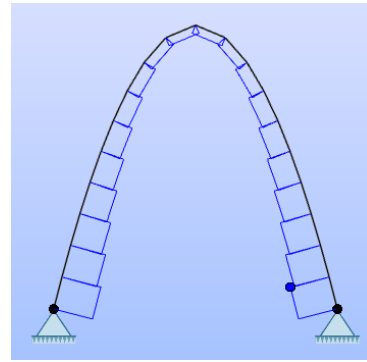
Dessuten har man registrert at tilsvarende logikk som for linjelaster brukes i dimensjonering av stålkonstruksjoner mht. knekning av bjelke-søyleforbindelser [avsnitt 6.3][2]. Momentfaktoren  $C_{my}$  er den usikre faktoren her, og den må valideres.

### 4.5.3 Parabelkoordinater

Følgende er en undersøkelse av beregningen av avstanden fra *joint* til node, langs *parabolic arch members*. Beregningene fra **fap2D** sammenlignes med analytiske beregninger gjort med det nettbaserte verktøyet WolframAlpha[18].



Figur 4.11: *Parabolic arch member* med  $h=2000$  mm.



Figur 4.12: Elementinndeling av modellen i figur 4.11.

Fremgangsmåte: Avstanden til nodene (nummerert fra og med venstre *joint*) i **fap2D** ble sammenlignet med tall utregnet av WolframAlpha. Input i WolframAlpha var kurvelengden fra  $x = -2000\text{mm}$  (posisjonen til venstre *joint*) til x-koordinatene for et utvalg noder (hentet fra **fap2D**). Resultatene er vist i tabell 4.1.

Modell: *parabolic arch member*

- *Joint* 1:  $(-2000, 0)$  [mm]
- *Joint* 2:  $(2000, 0)$  [mm]
- Høyde: 4000 mm
- Antall elementer: 20

Analytisk beregning:

- Program: WolframAlpha.com
- Parabeldefinisjon:  $f(x) = 4000 - x^2$
- Programinput: path length  $(4000 - x^2)$  from  $-2000$  to ...

Tabell 4.1: Resultatene av sti-lengdene som er beregnet.

Node-nummer	X-koord. For node [m]	Analytisk lengde [mm]	sti- Sti-lengde <b>fap2D</b> før [mm]	Sti-lengde <b>fap2D</b> etter [mm]
3	-1.7612162584516362	929,35679	929,36	929,36
5	-1.4878602024888068	1858,71354	1858,73	1858,71
12	0.41963349859271792	5111,46211	5112,14	5111,46
15	1.1595564275309398	6505,49725	6505,52	6505,50
19	1.7612162584516362	8364,21074	8364,21	8364,21

Før endringen var feilen størst langs toppen. Etter endring er sti-lengden korrekt avrundet.

Feilen var at sti-lengden ble utregnet nøyaktig til midten av elementet, og deretter ble halve lengden av det rette elementet trukket vekk eller lagt til for å finne sti-lengden til nodene. Dette var åpenbart feil, og er rettet til å ta hensyn til krumningen av *arch members*.

#### 4.5.4 Validering av eksentrisiteter

Undersøkelse av implementasjonen for eksentrisiteter.

Eksentrisiteter er definert på to måter:

- *MoveMember* (flytter enden av en *member* i forhold til et *joint*)
- *MoveJoint* (flytter et *joint* i forhold til én eller flere ender av *members*)

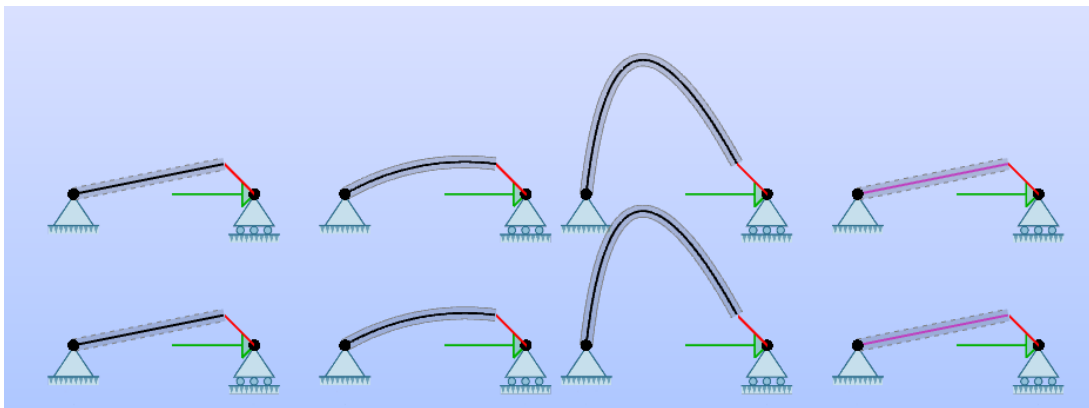
Dokumentet tar for seg to tester:

1. ulike kombinasjoner av eksentrisiteter for ulike typer *members*
2. håndtering av ugyldig endring av geometri for *circular arch members*

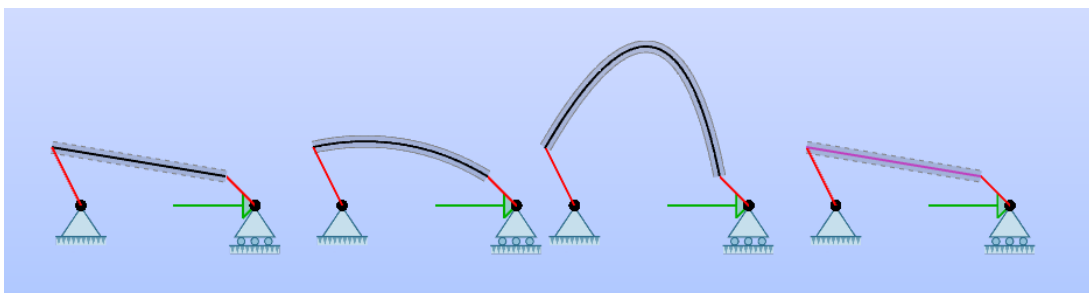
### Kombinasjoner

Ulike kombinasjoner av begge typene eksentrisiteter sammenlignes for å kvalitativt bekrefte at de oppfører seg som forventet. *Members* og *joints* er orientert slik at de kan sammenlignes lettere. Der *MoveMember* er benyttet er eksentrisitetene definert ved at *member*-en flyttes ( $Ecc_X, Ecc_Z$ ) = (-500mm, 500mm). Der *MoveJoint* er benyttet er eksentrisitetene definert ved at *joint*-ene flyttes ( $Ecc_X, Ecc_Z$ ) = (500mm, -500mm). Modellene vist i figur 4.13 sammenligner ulike *members* med ulike eksentrisiteter. Modellene i figur 4.14 sammenligner ulike kombinasjoner av *members* og eksentrisiteter.

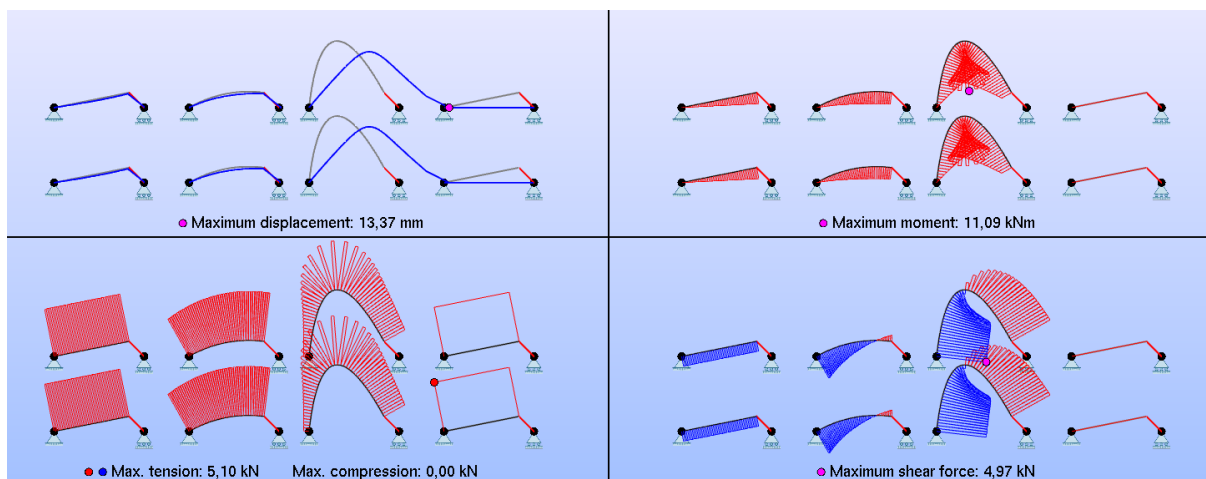
Resultatene som er vist i figur 4.15 og 4.16 viser at de ulike typene *members* med eksentrisiteter oppfører seg på lignende måte. Alle *members* takler begge typer eksentrisiteter, både i én og to ender, og ulike kombinasjoner av dem.



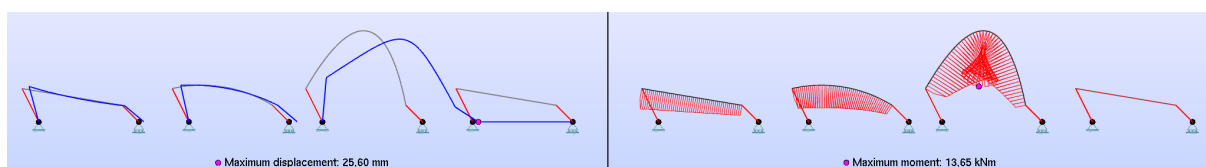
Figur 4.13: Én type eksentrisitet på ene siden av *beam*, *circular arch*, *parabolic arch*, og *bar members*. Øverste rad har eksentrisiteter av typen *MoveMember*, nederste rad har *MoveJoint*.



Figur 4.14: Eksempel på modeller med eksentrisiteter. I forsøkene er alle kombinasjoner av *member*-typer, eksentrisitetstyper, og *joint*/eksentrisitet-plassering forsøkt.



Figur 4.15: Resultater for modellene i figur 4.13.

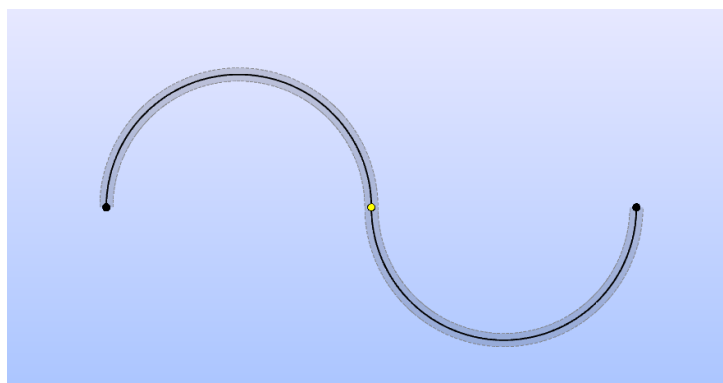


Figur 4.16: Resultater for modellene i figur 4.14. Alle kombinasjoner av eksentrisitetstyper, og *joint*/eksentrisitet-plassering hadde like resultater for hver type *member*.

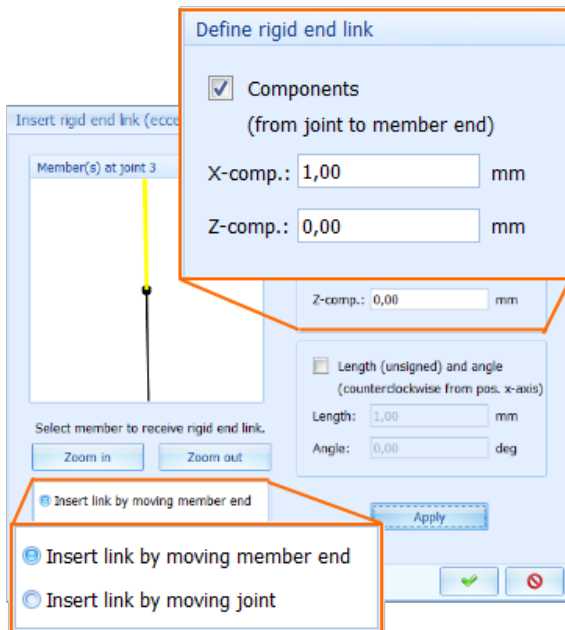
### Håndtering av geometri

Håndtering av eksentrisiteter som forsøker å endre *circular arch members* til ugyldig geometri, dvs. at  $L_{korde} > 2 \cdot radius$ , der  $L_{korde}$  er avstanden mellom de *joints* som definerer *member*-en.

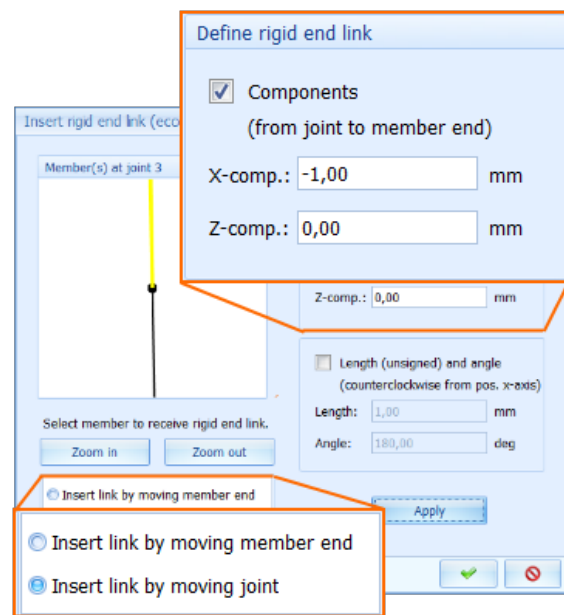
Modellen består av to *circular arch members* som er knyttet sammen i et *joint*. Det blir forsøkt å legge inn en *MoveMember* som flytter enden av venstre *arch member* og gjør den ugyldig ved at  $L_{korde} = 4001\text{mm}$ . Tilsvarende blir det forsøkt å legge inn en *MoveJoint* som flytter *joint*-et og gjør høyre *arch member* ugyldig. Programmet gir an advarsel om at flyttingen er ugyldig, og derfor ikke blir gjennomført.



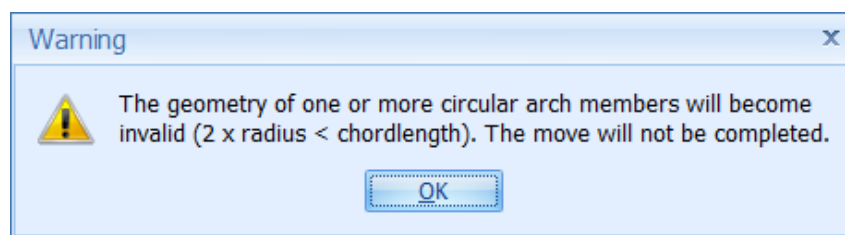
Figur 4.17: Modell av to *circular arch members* ( $r = 2000\text{ mm}$  og  $r = -2000\text{ mm}$ ). Eksentrisitet forsøkes satt inn i *joint*-et markert med gult.



Figur 4.18: Flytting av venstre *member* 1 mm i forhold til *joint*-et.



Figur 4.19: Flytting av *joint*-et 1 mm i forhold til venstre *member*.



Figur 4.20: Advarsel som dukker opp for begge tilfellene vist i figur 4.18 og 4.19.

## 4.6 Dokumentasjon for utviklere

Det er laget ny dokumentasjon som kan hjelpe fremtidige utviklere til å komme raskere inn i koden. Dessuten har dokumentasjon fra tidligere utviklere blitt revidert og utvidet i år, slik at den er oppdatert. All dokumentasjonen er blitt overlevert til veileder, slik at kunnskapen kan bringes videre i tråd med avsnitt 4.4.

### 4.6.1 Kommentarer i kildekoden

Som nevnt i avsnitt 4.4 er kildekoden den viktigste formen for kunnskapsdeling mellom utviklerne i prosjektet **fap2D**, siden revisjonskontrollen fornyes hvert år. Deler av kildekoden er dessverre tynt kommentert. Dessuten er deler av kommentarene utdaterte og stemmer ikke lenger overens med de faktiske kodelinjene de tilhører.

Et eksempel på en klasse med lite kommentarer er `FrmModel` i prosjektmappen `fap2D.Gui`. Denne klassen styrer store deler av modelleringsprosessen, blant annet vha. *mouse events*,

og er derfor svært viktig. I løpet av masteroppgaven har det blitt gjort mange endringer i `FrmModel` og det har blitt brukt unødvendig mye tid på å sette seg inn i metodene i klassen.

Derfor har det blitt lagt til mer detaljerte kommentarer på de stedene det har blitt utført endringer, slik at nye utviklere kan se tanken bak implementasjonene.

Det er også lagt til mer detaljerte oppsummeringer av metoder og *properties* som brukes, og mange metoder har fått mer presise navn.

### 4.6.2 Fargebruk i `fap2D`

Det kan være nyttig å ha oversikt over fargene som er brukt i grensesnittet til `fap2D`, siden det begynner å bli mange av dem. Fargene er et viktig virkemiddel for å gjøre GUI oversiktlig for brukeren.

Pedersen dokumenterte i sin masteroppgave hvordan farger er brukt, gjennom avsnitt 5.1 og 5.2 i [9]. Det har blitt laget et dokument som er basert på deler av innholdet i disse kapitlene, og som også tar for seg all den nye fargebruken som er innført i versjon **3.1**.

Dokumentet finnes i vedlegg B.2.6 Konklusjon.

### 4.6.3 *Mouse Events*

*Mouse events* er hendelser som omhandler musepekeren. I `fap2D` er dette begrenset til å gjelde pekerens *events* (handler) inne i modelleringsvinduet. Disse handlingene styres av klassen `FrmModel`. Dokumentet *Mouse events* i vedlegg B.2.6 Konklusjon er ment å gi en oversikt over alle disse *events* og tilhørende programlogikk.

Dokumentet introduserer hvilke *events* som benyttes, hvilke egenskaper som beskriver tilstanden til musepekeren, og rekkefølgen av *events*. Det er spesielt viktig å være oppmerksom på rekkefølgen av *events* fordi de kan skje tilnærmet samtidig. For eksempel vil både `MouseDown` og `MouseClicked` *fyes av* umiddelbart når bruker klikker på en av knappene på musepekeren.

Dokumentet ble opprettet for å være en referanse i sammenheng med *debugging* av modellering, men også for å vurdere om *mouse events* trengte en overhaling. Siden de fungerer akseptabelt er dette ikke nødvendig på nåværende tidspunkt.

### 4.6.4 *GridView Events*

Det er innført validering ved flytting av *joints* og *members*. Valideringen skal hjelpe brukeren slik at det ikke lages ugyldige eller tvilsomme konstruksjonsmodeller. Dessuten er valideringen nødvendig for å unngå *unhandled exceptions* som kan føre til programstopp.

I forbindelse med valideringen var det utfordrende å forstå den logiske koblingen mellom *joint data* i *left panel* og konstruksjonsmodellen i *modelling panel*. Endringer som gjøres i *joint data* påvirker nemlig konstruksjonsmodellen og vice versa.

*Joint data* er puttet inn i en `GridView`-instans. Det tok tid å lære seg å kontrollere de mange *events* knyttet til `GridView`, slik at man kunne få laget en robust valideringsprosess. Derfor



er det laget et dokument som forklarer hvordan disse *events* er brukt i **fap2D**, slik at denne kunnskapen kan overføres til nye utviklere.

Dokumentet er å finne i vedlegg B.2.6 Konklusjon. Det er skrevet på engelsk, i tråd med kildekode til **fap2D**.

#### 4.6.5 Reviderte *tooltips*

Det er laget et dokument, i vedlegg B.2.6 Konklusjon, som viser alle *tooltips* som er endret og lagt til i programmet. Det er praktisk å ha en oversikt slik at det er lettere å vedlikeholde informasjonen i dem.

*Tooltips* er nemlig brukt mer aktivt i versjon **3.1** enn før, for å erstatte deler av behovet for *tutorial* og lette behovet for å lese i UM. Derfor er de relativt detaljerte og spesifikke, og det kreves en større innsats for å vedlikeholde dem slik at de stemmer overens med UM og programmet.

#### 4.6.6 Installasjon og konfigurasjon av programvare

I prosjektet **fap2D** har det blitt laget dokumenter som beskriver hvordan utviklere får installert og konfigurert programvare for å drive utvikling. Disse dokumentene har blitt revidert og presisert årlig.

I hovedsak dreier det seg om et dokument som beskriver installasjon og oppsett av en server som kan brukes til versjonskontroll av koden. Dokumentet har blitt revidert i denne masteroppgaven, men endringene er relativt uinteressante for andre enn utviklerne og presenteres ikke. Den forrige versjon av dokumentet kan leses i [6, vedlegg F. SVN].

### 4.7 Gjenstående oppgaver

Det gjenstår fortsatt oppgaver som burde vært gjennomført. Flere av oppgavene stammer fra våren 2013 og enda lenger tilbake.

Enkelt oppgaver kunne vært gjennomført, men ut i fra prioriteringene som er gjort ble det ikke tid igjen. Det ville vært risikabelt å ta for seg nye oppgaver i slutfasen, siden det ikke ville blitt tilstrekkelig med tid til testing etterpå. Kompetansen i utviklerteamet har også satt begrensninger ved noen anledninger.

En del oppgaver ble vurdert, men har blitt lagt til side fordi man underveis fant ut at de ikke var nødvendige. Noen av vurderingene er gjort på bakgrunn av evalueringen av betatestingen. Dessuten har kost- og nyttevurderinger vist at en porsjon av oppgavene ikke er verdt å gjøre, med tanke på hva de vil tilføre brukerne.

Feil som ikke har vært mulige å reprodusere nevnes til slutt.

## 4.7.1 Oppgaver som bør gjennomføres

### 4.7.1.1 Systematisert validering

Validering av brukerinput og modelleringsprosessen har blitt forsøkt systematisert, men det er fortsatt ikke alle områder som valideres like nøye. Spesielt dialogboksene knyttet til *analysis ribbon page*. Praksisen med å benytte Masks fra avsnitt 3.10 bør videreføres for å få bedre kontroll over situasjonen.

### 4.7.1.2 Convert to LC

Som nevnt i punkt 3.5.2 fungerer ikke funksjonaliteten som konverterer laststeg fra *real time* og *influence line analyses* korrekt. Det er gjort tiltak for å begrense skaden ved å informere brukeren om hvordan feil kan unngås, men i grunnen bør funksjonaliteten rettes. Det ble ikke nok tid til dette.

### 4.7.1.3 Overgang fra Tao til OpenTK

For å forlenge levetiden til **fap2D** bør det utdaterte grafikkrammeverket Tao [21] byttes ut med sin erstatte - OpenTK [20]. Dessverre viste det seg at man ikke hadde den nødvendige kompetansen for å gjøre dette, siden endringene var mer drastiske enn antatt. Pedersen [9, avsnitt 6.6 og vedlegg A] diskuterer rammeverket for tegning i .

### 4.7.1.4 Videreutvikling av zoom

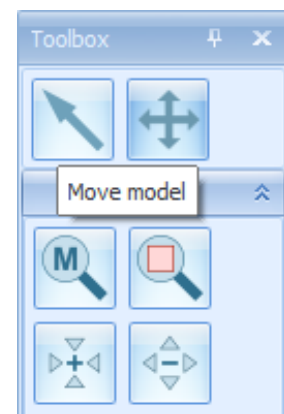
Zoom-funksjonaliteten kan videreutvikles på noen punkter. Det skal nevnes at ifølge evaluering av betatesting var det få som hadde problemer med å zoome. Se vedlegg B for detaljer.

#### Zoom til musepekerens posisjon

Når brukeren zoomer vha. hjulet på musen vil det alltid zoomes i forhold til punktet som for øyeblikket er midt i *modelling panel*. En alternativ praksis er at hjulet zoomer om musepekerens posisjon på skjermen. En slik zoom-definisjon gjør at zooming kan utføre en kombinasjon av flere oppgaver (i figur 4.21) samtidig. Det fører til færre tastetrykk og bedre flyt i operasjonene. Det antas at kostnaden for å implementere dette er lav i forhold til nytteverdien, og at det derfor allerede burde vært gjort.

#### Zoom-rekkevidde

Rekkevidden for zooming er avhengig av hvilken *grid spacing* som brukes. Dvs. at man må endre *grid spacing* før man kan zoome lengre inn eller ut. Så lenge *grid*-et er fornuftig

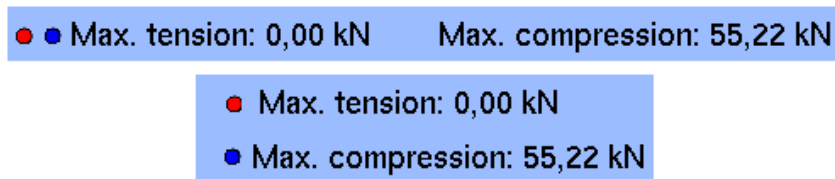


Figur 4.21: Verktøy som kan kombineres i en bedre zoom-funksjon.

definert vil ikke brukeren merke noe til dette, men viss ikke vil det være vanskelig å forstå. Problemstillingen oppstår sjelden så den er ikke spesielt viktig.

#### 4.7.1.5 Tekst forsvinner fra *modelling panel*

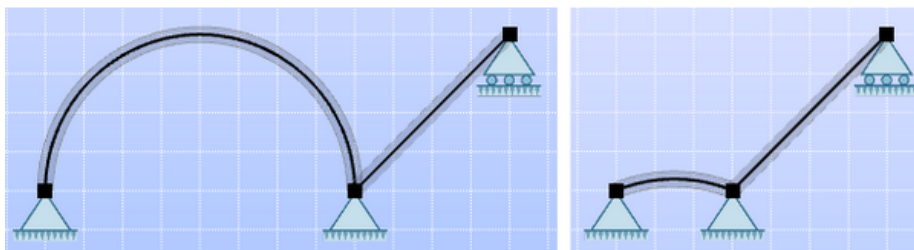
Tekst som vises i *modelling panel* kan forsvinne, antagelig av to grunner. Den ene er lav skjermopløsning som gjør teksten ikke får plass i panelet. Den andre grunnen er antagelig at tegnefunksjonene som brukes er basert på OpenGL 2.1. Grafikkort som ikke støtter versjon 2.1 vil ikke kunne tegne teksten i panelet. UM påpeker at OpenGL 2.1 er et krav så det må brukeren forholde seg til. Det stilles også krav til en viss skjermopløsning<sup>4</sup>. Figur 4.22 illustrerer hvordan linjeskift kan brukes for å gjøre teksten smalere. Fontstørrelsen kunne dessuten vært basert på skjermopløsningen i stedet for å være konstant.



Figur 4.22: Illustrasjon av smalere tekst vha. linjeskift.

#### 4.7.1.6 Flytting av systemer med *arch members* kan endre geometrien

Dersom en konstruksjon markeres og flyttes kan geometrien til *arch members* endres. Brukeren flytter ikke så ofte konstruksjonen på denne måten, men feilen burde vært rettet for å skape tillit til *arch members* hos brukeren.

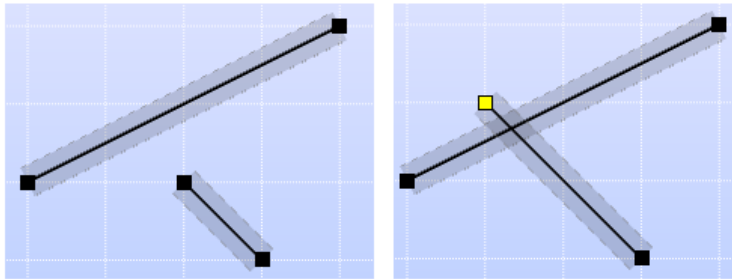


Figur 4.23: Modellen før flytting (t.v) og etter (t.h).

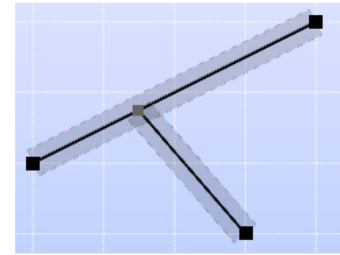
#### 4.7.1.7 *Snap joint til member*

Ved å starte eller avslutte modelleringen av en ny *member* nært et punkt langs en eksisterende *member* snapper den nye *memberen* til et *internal joint* på den eksisterende *memberen*. Et nytt *internal joint* opprettes automatisk i punktet hvis det ikke finnes et fra før av. Hvis man derimot flytter enden av en eksisterende *member* til et punkt i nærheten av en annen *eksisterende member* vil det ikke *snappes* til *memberen*, det vil *snappes* til rutenettet.

<sup>4</sup>Se vedlegg F.1.



Figur 4.24: Uønsket oppførsel: Før flytting (t.v) og etter flytting (t.h.)



Figur 4.25: Ønsket oppførsel: *Snapper* til *member* via *internal joint*.

#### 4.7.1.8 Loggfil skrives til programfiler

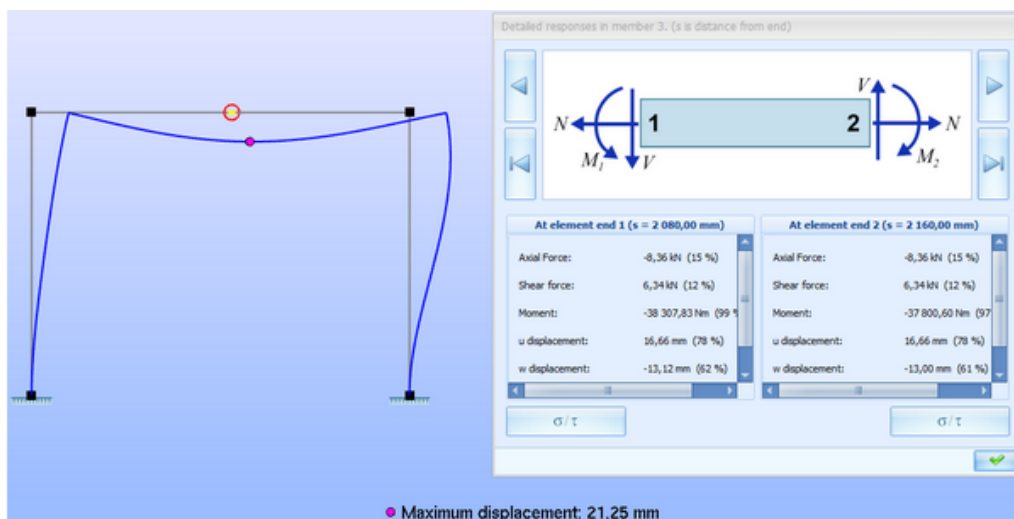
Loggfilen som skrives ut når en analyse kjøres blir skrevet ut til mappen programfiler-mappen der **fap2D** ble installert. Problemet med å skrive loggfilen til programfilene er at enkelte brukere ikke har skriverettigheter i disse mappene. I forbindelse med betatestingen opplevde man at PC-er på datasaler ved Lerkendalsbygget ikke fikk kjørt analyser pga. manglende skriverettigheter. Dermed ble programmet ubrukelig.

Det er beregningskjernen **Frame2D** som bestemmer plasseringen til loggfilen. Veileder påpeker at det blir vanskelig å få endret plasseringen på kort sikt. Planen var egentlig å skrive ut loggfilen til en mappe som er knyttet til PC-brukeren i stedet, for å sikre at det er skriverettigheter i alle tilfeller.

#### 4.7.1.9 Visning av modell og elementresultater samtidig

Når man trykker på et element for å få fram dialogen med elementresultater blir modellen som regel gjemt bak denne dialogen. Det ville vært fint å vise modellen ved siden for dialogen i stedet slik at man får sett begge samtidig. Dette er spesielt nyttig i sammenheng med den nye funksjonaliteten "Navigering av elementresultater" i avsnitt 3.6.5. For å implementere endringen burde man erstattet dialogen med et panel som kan stilles ved siden av *modelling panel* når man trykker på et element. Se figur 4.26 for en forenklet illustrasjon av løsningen (uten panel).

*Vurdering: Endringen ses på som en veldig liten forbedring i forhold til arbeidet som må gjennomføres for å lage det nye panelet. Dessuten er det ikke vanskelig for brukeren å dra dialogen til siden slik at modellen synes ved siden av.*



Figur 4.26: Illustrasjon: Vise modell og elementresultater i hvert sitt panel.

#### 4.7.1.10 Member data-tabellen

*Member data*-tabellen i *left panel* er ikke mye brukt fordi det ikke er mulig å endre *members* vha. den, bare se informasjon om den. Endringer for *members* kan gjøres vha. *member properties dialog*, så det er ikke kritisk at endringer skal kunne bli gjort i *member data*-tabellen. Designet og innholdet i tabellen kan forbedres slik at den blir mer interessant å bruke, men den kunne like gjerne vært fjernet fra **fap2D**. Problemstillingen er nevnt i [9, avsnitt 6.7].

Id	Joint End 1	Joint End 2	Material
1	1	2	Steel

Sub Members

Search: Cross section ... Cross section ...

Sub Members: SHS 100x100x10 SHS 100x100x10

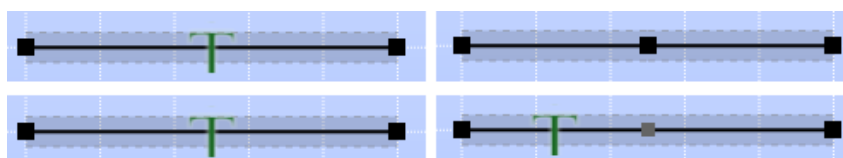
Figur 4.27: *Member data*-tabellen.

#### 4.7.1.11 Overføring av *properties* ved deling av *members*

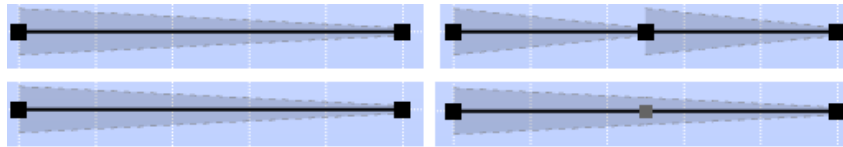
Ved deling av *members* i flere *members* blir ikke alle *properties* overført til de nye *members*: *Line loads*, *initial strain/temperature*.

Ved deling av *members* i *submembers* blir ikke alle *properties* overført til alle nye *submembers*: *Initial strain/temperature*.

Det bør defineres bedre regler for overføringen av egenskaper. Klassen som i hovedsak må endres er `FrmMemberPropertiesDialog`.



Figur 4.28: Deling i nye *members* (øverst) - Deling i *submembers* (nederst). Initialtøyning blir ikke overført.



Figur 4.29: Deling i nye *members* (øverst). Deling i *submembers* (nederst).  
*Cross section* blir ikke overført.

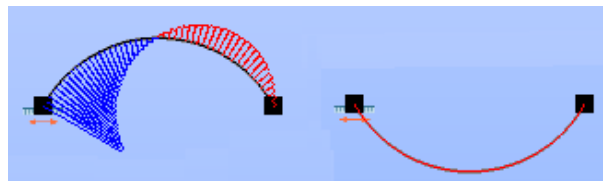
#### 4.7.1.12 Bruk av .resx-filer i stedet for mappe med ressurser

Aasmundrud [6, kap. 7] har nevnt at ressurser i “Resources”-mappen bør legges i interne .resx-filer i Visual Studio. Grunnen er at det forbedrer ytelsen i programmet og at filene skjules for bruker ved å legges til i .dll-filer. Dette kunne godt vært gjort dersom det forbedrer ytelsen. Vi er usikre på hvor mye ytelsen kan bedres siden vi ikke har tatt tak i problemstillingen. Skjuling av ressursfiler for brukeren ses ikke på som nødvendig.

#### 4.7.1.13 Earthquake analyses

##### **Earthquake response spectrum analysis av mer enn én konstruksjon samtidig**

Ut i fra resultatdiagrammene som vises ved *response spectrum analysis* virker det som om bare én sammenhengende konstruksjon kan analyseres av gangen, se figur 4.30. Det er ikke ofte man har behov for å analysere flere konstruksjoner samtidig, men det kan være nyttig i og med at man da kan sammenligne konstruksjoner veldig oversiktlig og enkelt. Likevel har ikke denne feilen blitt prioritert siden man kan sammenligne modeller ved å kjøre analyse av to ulike modellfiler hver for seg og sammenligne etterpå.



Figur 4.30: Bare den ene av to konstruksjoner viser respons.

#### **Bedre import/eksport av jordskjelv-data**

Aasmundrud [6, kap. 7] har nevnt at det bør bli lagt til flere filformater for eksport og import av jordskjelv-data. Import av jordskjelv er nemlig veldig sensitiv for formateringen i datafilen. Dette har ikke blitt prioritert fordi jordskjelvanalyser er ment for viderekomne brukere og har blitt nedprioritert til fordel for modifikasjoner knyttet til basisfunksjonalitet og enklere analysetyper.

#### 4.7.1.14 Kopiering av *structural items* og generering av malfiler

To tiltak som kan gjøre det enklere å lage store modeller og gjenbruke *structural items* er kopiering og malfiler. Begge deler er tidligere diskutert i [10, avsnitt 4.2 og 4.3] og [9, avsnitt 6.9]. Man antar at det vil kreve en del arbeid å gjennomføre tiltakene og dessuten fungerer *grid* fint for å plassere ut nye *items* hurtig. Ikke høyt prioritert.

#### 4.7.1.15 Knytte lastbaner til beregningsmodell

I kildekoden er *load path*-objektene knyttet til konstruksjonsmodellen. De burde heller vært knyttet til beregningsmodellen ifølge Erstad[8, avsnitt 8.5], siden det er mest naturlig. Arbeidet vil dog gi liten gevinst for brukeren og ble av den grunn nedprioritert.

#### 4.7.1.16 Tilleggsfunksjonalitet for jordskjelvanalyser

Aasmundrud [6, kap. 7] nevner flere implementasjoner som kunne vært lagt til for jordskjelvanalysene. De har blitt oversett fordi flere av endringene krever utvidelser i **Frame2D** og man har prioritert mer grunnleggende funksjonalitet.

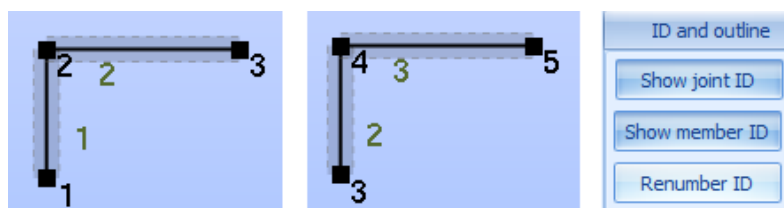
#### 4.7.1.17 Mer avansert fjerning av numerisk støy

Aasmundrud implementerte en *setting* for å fjerne numerisk støy [6, kapittel 7]. Metoden som brukes er visstnok primitiv og brukeren bør kunne styre fjerningen mer nøyaktig. Krever utvidelser i **Frame2D**.

### 4.7.2 Oppgaver som kan ses bort fra

#### 4.7.2.1 Undo gir ikke tilbake de gamle *joint*- og *member*-numrene

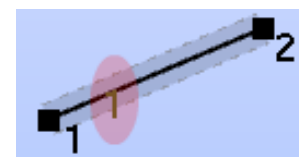
Dersom *undo*-funksjonen brukes skal endringer som er gjort tilbake tilles til forrige steg i *modelling panel*. I stedet for å gi tilbake gamle *joint*- og *member*-numre så blir det i stedet laget nye numre. Oppførselen kan være forvirrende for brukeren, men siden det er mulig å renummerere *joints* og *members* via *toolbox panel* er det ikke så viktig å ordne dette.



Figur 4.31: Fv: Før *undo* - Etter *undo* - Verktøy for renummerering i *toolbox panel*.

#### 4.7.2.2 Tegning av *joint*- og *member*-numre

Metoden som bestemmer hvor *joint*- og *member*-numre skal tegnes er for primitiv. Den tar ikke hensyn til at *members* kan ha ulike vinkler, og dersom flere *joints* eller *members* overlapper hverandre tegnes numrene oppå hverandre. Pedersen [9, avsnitt 6.12] nevner at tegningen ble utbedret våren 2013, og at den bør utbedres mer. Feilen er kosmetisk og lite viktig for brukeren.



Figur 4.32: *Member*-nummer synes nesten ikke.

### 4.7.2.3 Bruk av venstre musetast i resultatvisningen

Venstre musetast brukes ikke i resultatvisningen, mens høyre musetast brukes for å velge ut elementer og vise dialogbokser med resultater. Veileder argumenterte for at høyretasten generelt er “informasjonstasten” i **fap2D**, mens venstretasten er “operasjonstasten”. Denne logikken gir mening, men man kan også argumentere for at venstretasten er “primærtasten”, mens høyre er “sekundærtasten” og bare skal brukes når venstre alt er i bruk. I evalueringen av betatestingen kom det frem at bruk av høyretasten er logisk<sup>5</sup> og derfor er det ikke behov for endring av praksis.

### 4.7.3 Feil som ikke er gjenskapt

Feilene i dette avsnittet har blitt oppdaget, men ikke gjenskapt nok ganger til at man har vært i stand til å rette dem. De to første feilene som nevnes er relativt grove dersom de er reelle, mens resten anses som småpirk.

Sent i masteroppgaven ble det oppdaget at laster ikke alltid blir oppdatert. Av og til når man endrer størrelsen på eksterne laster blir ikke endringene registrert i beregningsmodellen. Det fører til at uendrede resultater vises selv om brukeren faktisk endret lastene. Ettersom det ikke har blitt tid til å finne årsaken kan man ikke si annet enn at det er kritisk å rette den opp. Feilen ble oppdaget i både versjon **3.0** og **3.1**.

Vit.ass. Holmstrøm rapporterte en feil angående *coupling springs*. Modeller med rotasjonsstivhet i form av fjærer ga ikke samme respons første og andre gang en analyse har ble gjennomført. Etter andre kjøring var responsen stabil. Det har vært blitt gjort flere rettinger av feil knyttet til rotasjonsfjærer.<sup>6</sup> Som følge av det kan feilen være rettet, men det er ikke helt sikkert.

Noen brukere har opplevd treg og lite responsiv oppførsel ved veksling mellom bruk av *modelling* og *results ribbon pages*. Andre har opplevd det samme når de har hatt flere modellfiler åpne samtidig i programmet. Det har blitt gjort enkle forsøk på å teste minnebruken, men forbruket er sjelden mer enn 150 MB data.

Høsten 2013 opplevde man at endring av opplagerbetingelser ikke fungerte som ventet [10, avsnitt 3.4.5].

Det er tidligere blitt varslet om at *modelling panel* kan bli hvitt når lokal akse eller *point damper* legges inn i en modell. Rapporter om dette er ikke registrert etter våren 2013.

Utviklerteamet har opplevd å få feilmelding når en *member* har blitt plassert på en *member* som har fordelt last knyttet til seg. Årsaken er uviss.

---

<sup>5</sup>Se vedlegg B.2.5.1 *Resultater fra evaluering* for detaljer.

<sup>6</sup>Se vedlegg A.4.16 og A.4.17.



# Kapittel 5

## Konklusjon og videre arbeid

Dette kapitlet presenterer konklusjoner fra arbeidet som er utført og gir anbefalinger til videre arbeid med prosjektet og programmet **fap2D**.

### 5.1 Konklusjon

De fleste av målene som ble satt for arbeidet i avsnitt 1.1 regnes som oppnådde. For eksempel er tegning, markering og presentasjon av *structural items* og analyseresultater tydelig oppgradert. Et bredt spekter av både lett synlige og mer skjulte feil har blitt rettet, og revisjon av *settings* og *User's Manual* har gjort programmet mer brukervennlig. Det er som nevnt vanskelig å kvantifisere graden av måloppnåelse, men antallet forbedringer som er utført indikerer at mye er oppnådd.

Tilstandsrapporten viser at det fortsatt gjenstår en del oppgaver, blant annet knyttet til lisensiering, installasjon og grundigere testing. Dersom tiden hadde strukket til skulle man gjerne utført flere oppgaver og endret prioriteringene undervegs, men det var til tider krevende å se det store bildet.

Likevel legges det vekt på at erfaringene i betatestingen var positive og de ytterligere forbedringene utført i versjon **3.1** gjør oss trygge på at **fap2D** kan ta steget ut av betafasen og stå på egne ben i årene fremover.

## 5.2 Anbefalinger for videre arbeid

Dersom **fap2D** videreføres som et langsiktig prosjekt bør man etablere et kontinuerlig system for versjonskontroll med integrert *bug system* slik at nye utviklere ikke må starte fra grunnen av hvert år. Videreføringen av kunnskap som har blitt gjort frem til nå er nemlig ikke utpreget effektiv. Mye kunnskap gått tapt fordi det ikke eksisterer detaljert nok dokumentasjon for valg som er tatt i løpet av utviklingen.

Siden **fap2D** er klargjort for å stå på stødige bein bør innsatsen i prosjektet **fap2D** rettes mot å følge opp og promotere programmet for bruk i undervisning ved fag tilknyttet Institutt for konstruksjonsteknikk.

Ellers i det videre arbeidet er det naturlig å vurdere punktene i avsnitt 4.7. Vi vil trekke ut noen punkter som det bør fokuseres spesielt på:

- Det grafiske rammeverket Tao Framework som benyttes i **fap2D** er utdatert og avløst av OpenTK. Oppdatering av tegnemethodene og overgang til ny teknologi anbefales for forlenge levetiden til programmet.
- Zoom bør videreutvikles for at det skal være enklere å forholde seg til modeller av ulik størrelse, og for å utnytte musehjulet maksimalt.
- Ved deling av *members* bør tilhørende egenskaper overføres til de nyopprettede *members* på en veldefinert måte.
- Verktøy for kopiering/klipping/liming er fremdeles fraværende. Selv om det ikke er akutt etterspørsel anses funksjonaliteten som nyttig.
- Lastbaner bør knyttes til beregningsmodell istedenfor konstruksjonsmodell.
- Konvertering av laststeg fra *real time/influence line analysis* til statisk lasttilfelle må kvalitetssikres for å unngå den uventede oppførselen som er nevnt i avsnitt 4.7.1.2.
- En rapportgenerator vil være et nyttig verktøy for at brukere enkelt skal kunne dokumentere arbeid utført i **fap2D**. Denne kan bygge videre på logikken som er benyttet for eksport.
- Arbeidet med å innføre strengere validering bør fortsettes.

# Referanser

- [1] Robert D. Cook, David S. Malkus, Michael E. Plesha, Robert J. Witt. *Concepts and applications of finite element analysis, 4th edition*. John Wiley og Sons, Inc., 2002.
- [2] Standard Norge. *Eurocode 3: Prosjektering av stålkonstruksjoner - Del 1-1: Allmenne regler og regler for bygninger*. 2005.
- [3] Brita Årvik, Daniel Aase. *Prosjektoppgave, GUI for fap2D*. Tekn. rapp. Trondheim: Institutt for konstruksjonsteknikk, NTNU, des. 2010.
- [4] Kolbein Bell. *Matrisestatikk, statiske beregninger av rammekonstruksjoner*. Trondheim: Tapir Akademiske Forlag, 2011.
- [5] Fredrik Larsen. "fap2D - Dimensjonering av komponenter av stål". Masteroppg. Trondheim: Institutt for konstruksjonsteknikk, NTNU, jun. 2011.
- [6] Erik Aasmundrud. "Implementering av jordskjelvberegninger i fap2D". Masteroppg. Trondheim: Institutt for konstruksjonsteknikk, NTNU, jun. 2013.
- [7] Kolbein Bell. *fap2D User's Manual*. Institutt for konstruksjonsteknikk, NTNU. Trondheim, mai 2013.
- [8] Frans Erstad. "Videreføring av lineær statisk analyse i programmet fap2D". Masteroppg. Trondheim: Institutt for konstruksjonsteknikk, NTNU, jun. 2013.
- [9] Kristian Pedersen. "Forbedring og kvalitetssikring av programmet fap2D". Masteroppg. Trondheim: Institutt for konstruksjonsteknikk, NTNU, jun. 2013.
- [10] Torjus Sandviken. *Prosjektoppgave, GUI for fap2D*. Tekn. rapp. Trondheim: Institutt for konstruksjonsteknikk, NTNU, des. 2013.
- [11] Espen Shin Skogsrud. *Prosjektoppgave, GUI for fap2D*. Tekn. rapp. Trondheim: Institutt for konstruksjonsteknikk, NTNU, des. 2013.
- [12] Focus Software AS. *Focus Konstruksjon*. Apr. 2014. URL: <http://focus.no/produkter/focus-konstruksjon.aspx>.
- [13] Microsoft Corporation. *Visual Studio*. Jun. 2014. URL: <http://www.visualstudio.com/>.
- [14] Microsoft Corporation. *Visual Studio 2013 - Code Metrics Values*. Jun. 2014. URL: <http://msdn.microsoft.com/en-us/library/bb385914.aspx>.
- [15] Microsoft Corporation. *Visual Studio Online*. Jun. 2014. URL: <http://www.visualstudio.com/en-us/products/what-is-visual-studio-online-vs.aspx>.
- [16] Tekla Corporation. *Tekla*. Jun. 2014. URL: <http://www.tekla.com/solutions/structural-engineers>.
- [17] Developer Express Inc. *DevExpress WinForms*. Jun. 2014. URL: <https://www.devexpress.com/Products/NET/Controls/WinForms/>.
- [18] Wolfram Alpha LLC. *WolframAlpha*. Jun. 2014. URL: <http://www.wolframalpha.com/>.

- [19] Dassault Systèmes S.A. *SIMULIA Abaqus*. Apr. 2014. URL: <http://www.3ds.com/products-services/simulia/portfolio/abaqus>.
- [20] Inc. SourceForge Dice Holdings. *The Open Toolkit library - OpenTK*. Jun. 2014. URL: <http://sourceforge.net/projects/opentk/>.
- [21] Inc. SourceForge Dice Holdings. *The Tao Framework*. Jun. 2014. URL: <http://sourceforge.net/projects/taoframework/>.
- [22] Simon Tatham. *PuTTY*. Jun. 2014. URL: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.
- [23] TortoiseSVN. *TortoiseSVN*. Jun. 2014. URL: <http://tortoisesvn.net/>.
- [24] VisualSVN. *VisualSVN*. Jun. 2014. URL: <http://www.visualsvn.com/visualsvn/>.
- [25] Wikipedia. *Bug tracking system*. Jun. 2014. URL: [http://en.wikipedia.org/wiki/Bug\\_tracking\\_system](http://en.wikipedia.org/wiki/Bug_tracking_system).

# Tillegg A

## *Release notes* - Versjonsmerknader

### Introduksjon

Endringene som er gjort mellom **fap2D 3.0.0 Beta** og **fap2D 3.1** er dokumentert i detalj i versjonsmerknadene. I løpet av våren 2014 har utviklerteamet produsert flere nye versjoner av **fap2D**, dog kun for internt bruk.

Ettersom de nye versjonene har blitt testet ut har det i visse tilfeller blitt konkludert med at endringene som ble gjort ikke fungerte optimalt. Derfor er i disse endringene blitt gjort om igjen i nyere versjoner. For å vise dette er alle endringer beskrevet med de interne versjonsnumrene. Endringer som ikke har blitt gjort om igjen senere er bare merket med versjonsnummer **3.1**.

### Versjonsnummerering

Tabell A.1: Versjonsnummerering.

Nummer	Beskrivelse
<b>3.0.0 Beta</b>	Sluppet i juni 2013.
<b>3.3.0 Developer Edition</b>	Intern versjon, mars 2014.
<b>3.5.0 Developer Edition</b>	Intern versjon, mai 2014.
<b>3.5.1 Developer Edition</b>	Intern versjon, med noen små endringer fra forrige versjon.
<b>3.1 (Release)</b>	Den nye versjonen, juni 2014.

De interne versjonene har fått høyere nummer enn versjonen som slippes til åpen bruk fordi vi valgte at 2. siffer skulle gjenspeile måneden da versjonen ble sluppet. Da vi til slutt skulle bestemme nummeret for den åpne versjonen bestemte vi oss for at programmet ikke er så drastisk endret at vi kunne kalle det for **fap2D 4.0**. Dermed hoppet vi tilbake.

Versjon **3.1** slippes som en ordinær utgivelse, til bruk ved Institutt for konstruksjonsteknikk, NTNU. Versjonsnummeret er økt fra **3.0** til **3.1** fordi det gjenspeiler at mange mindre, men viktige endringer er innført. Vi holder oss som sagt til samme hovedversjonsnummer som før fordi det ikke er lagt til vesentlige nye hovedfunksjoner (analysetyper).

Vi ser ikke behovet for å være så detaljerte at vi trenger et tredje siffer. Utviklingsprosjektet **fap2D** er tross alt veldig lite og oversiktlig. Siden betatesting regnes som gjennomført og bestått dropper vi også "**beta**" fra navnet.

## Klassifisering og beskrivelse av endringer

Tabell A.2 viser hvordan endringene er klassifisert.

Tabell A.2: Klassifisering av endringer.

Klassifisering	Beskrivelse
Feil	Retting av (åpenbare) feil.
Endring	Mindre justeringer av eksisterende funksjonalitet.
Nytt	Ny implementasjon som gir mer funksjonalitet i programmet.

Alle endringene har fått et unikt nummer i tillegg til klassifiseringen slik at det enkelt kan refereres til. For hvert punkt vises hvilke personer som har rapportert og gjennomført endringen og når dette har skjedd. Deretter beskrives tilstanden før og etter gjennomført endring, med figurer og tekst. Til slutt vises alle klassene (i programkoden) som har blitt endret.

## A.1 GUI

### A.1.1 Plassering av ribbon page groups og buttons i modellering ribbon page

ID	42
Klassifisering	Endring
Modifisert dato	12.02
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	November 2013
Rapportert av	Torjus

### Endringer

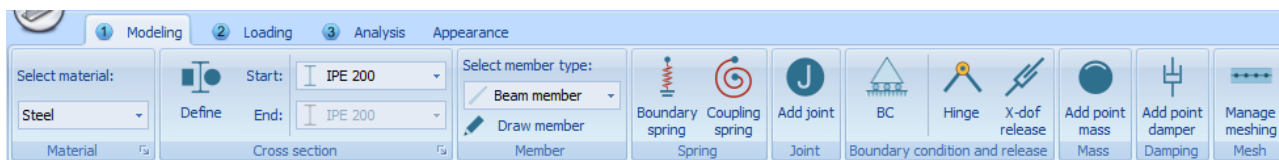
Endret rekkefølgen på *modelling items* i *Modelling ribbon* slik at de mer grunnleggende *items*, som brukes oftere, står lenger til venstre. Dette gjelder *Add joint* og *Boundary conditions*.

*Springs*, *damping* og *mass* er satt inntil hverandre fordi de ofte brukes sammen i dynamiske analyser.

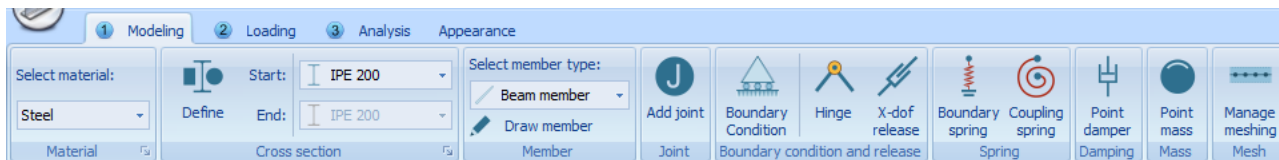
Endrede klasser:

- `FrmMainGui.designer`

<i>Namespace</i>	Endrede klasser
<code>fap2D.Gui</code>	<code>FrmMainGui.designer</code>



Figur A.1: Før endring.



Figur A.2: Etter endring.

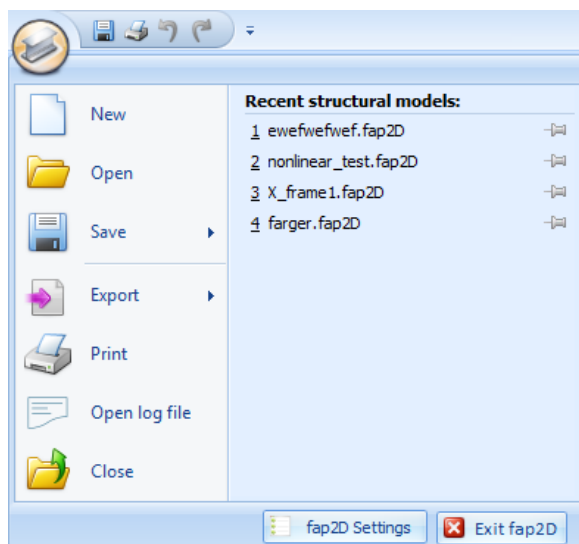
## A.1.2 Layout i *Application menu*

ID	70
Klassifisering	Endring
Modifisert dato	20.03
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	
Rapportert av	Torjus

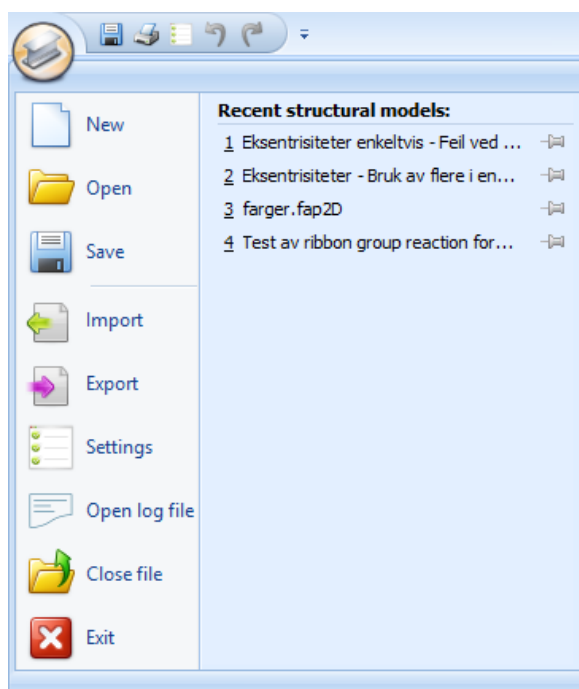
### Endringer

Flyttet knappene for *Settings* og *Exit* til den samme listen som de andre knappene i *Application menu*. Ryddigere og mer konsistent.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmMainGui.designer FrmMainGui.ApplicationMenu



Figur A.3: Før endring.



Figur A.4: Etter endring.

### A.1.3 *Settings button i Quick access toolbar*

ID	71
Klassifisering	Nytt
Modifisert dato	20.03
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	
Rapportert av	Torjus



## Endringer

Lagt til *Settings button* i *Quick access toolbar* for raskere tilgang til innstillingene.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmMainGui.designer



Figur A.5: Før endring.



Figur A.6: Etter endring.

### A.1.4 Bruk av DevExpress *controls* i stedet for Windows Forms *controls*

ID	32
Klassifisering	Feil
Modifisert dato	14.02
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	13.02
Rapportert av	Torjus

## Feil

Ved noen tilfeller er det brukt controls fra Windows Forms-rammeverket i stedet for DevExpress sin implementasjon av Windows Forms:

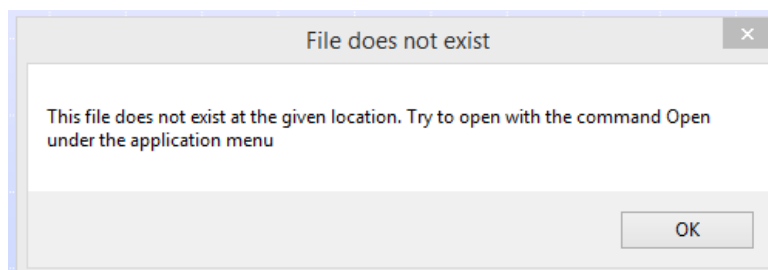
Windows Forms	DevExpress
MessageBox	XtraMessageBox
TextBox	TextEdit

Tabell A.3: *Controls*.

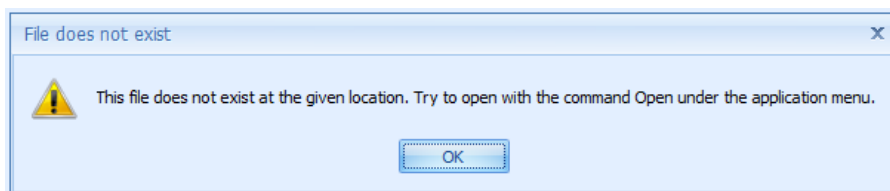
## Endringer

Endret til DevExpress-alternativene. De fleste elementene i brukergrensesnittet skal nå være basert på DevExpress.

<i>Namespace</i>	Endrede klasser
fap2D.Utilities	MultisampleGlControl
fap2D.Gui	FrmMainGui.RecentFiles
	FrmSumForcesInModel



Figur A.7: Før endring.



Figur A.8: Etter endring.

## Detaljer

Feilen er av kosmetisk art, men er rettet fordi det gjør *GUI* mer helhetlig. Siden rammeverket DevExpress er basisen for *GUI* bør det brukes så sant det kan. Likevel finnes det fortsatt Windows Forms *controls* der det ikke er fullgode DevExpress-alternativer, og der bruken av Windows Forms er lite synlig for brukeren.

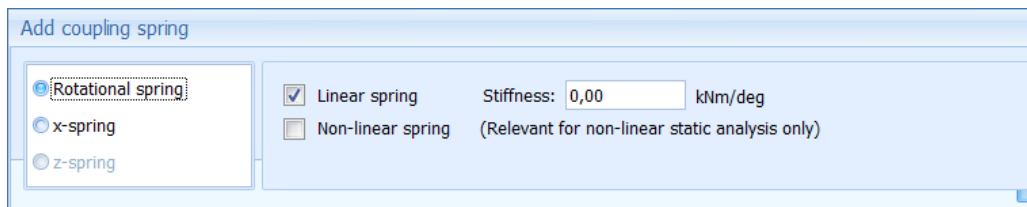
### A.1.5 Utseendet til dialoger for koblingsfjær og responsparameter

ID	17
Klassifisering	Endring
Modifisert dato	19.05
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	Høst 2013
Rapportert av	K. Bell

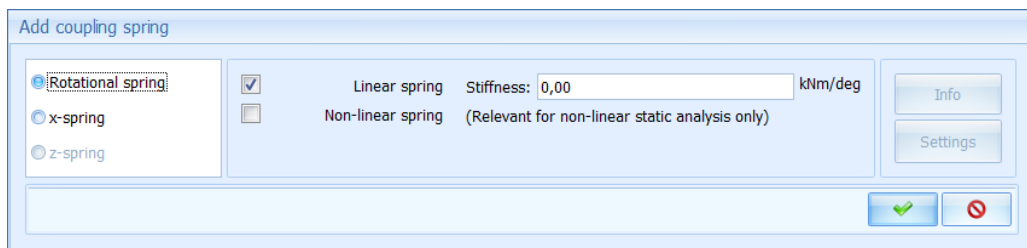
## Endringer

Endrer utseendet til enkelte dialogbokser som er spesielt preget av en feil der de ikke tilpasser seg innstillinger i **Windows** som øker størrelsen på tekst og symboler til 125% eller 150%. Spesielt dialogboks for *coupling spring/boundary spring*, og responsparameter fikk uventet og begrensende utseende.

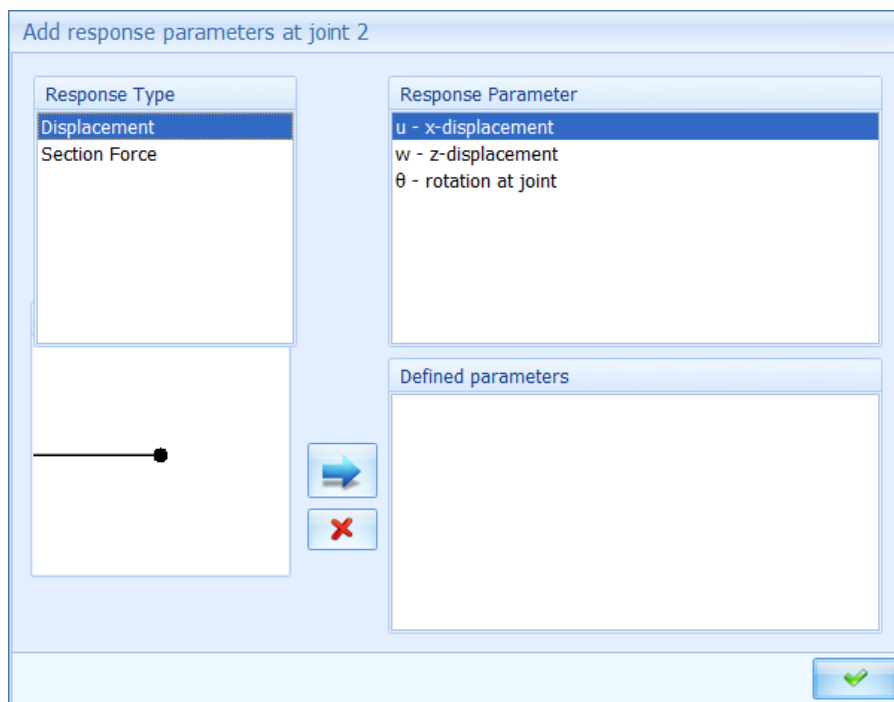
<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmSpringDialog.cs FrmSpringDialog.Designer.cs FrmResponseParameterDialog.cs FrmResponseParameterDialog.Designer.cs



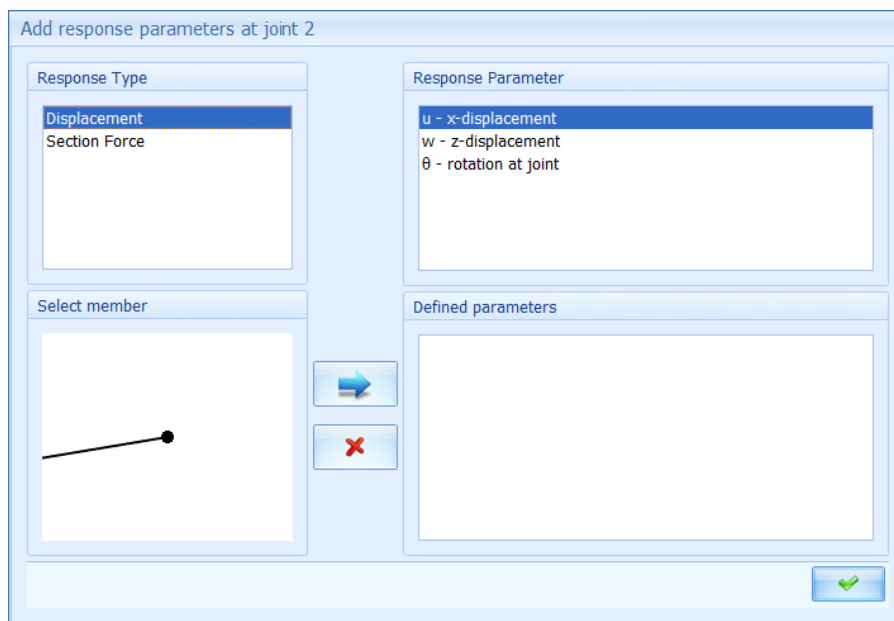
Figur A.9: Før endring - Dialog for fjær - Knapper for å godta/avbryte og informasjons-knapper er skjult utenfor dialogen.



Figur A.10: Etter - Dialog for fjær - Alle knappene er tilgjengelige.



Figur A.11: Før – Dialog for responsparameter - *User Control* for å velge *member* er delvis feilplassert.



Figur A.12: Etter - Dialog for responsparameter - Alle *controls* er justert.

## Detaljer

Løsningen var å benytte **DevExpress** sin `XtraLayoutControl`. `XtraLayoutControl` er utviklet for å enkelt og raskt håndtere layout til en dialog selv om nye kontroller legges til, eller dialogen endrer utseende.

Et utdrag fra **DevExpress** sin dokumentasjon:

“[...] After controls have been arranged, the `XtraLayoutControl` maintains a consistent layout by preventing the controls from being overlapped and misaligned even when the font settings are changed or the form is resized.”

[ <https://documentation.devexpress.com/#WindowsForms/CustomDocument3407/> ]

### A.1.6 Skjule knappene *Show resultants* og *Reaction forces*

#### Feil

Knappene *Resultants* og *Reaction forces* er deaktiverte for mange analysetyper så de kan like godt skjules for disse.

#### Endring

Knappene er flyttet til en egen *ribbon page group* og gruppen aktiveres kun for de analysene som har behov for å vise resultat- og reaksjonskrefter. Dvs. for lineær og ikkelineær statisk analyse, samt *real time analysis*.

#### Argumentasjon

Knappene er fjernet fordi de er unødvendige å vise. Bedre oversikt i GUI etterpå.

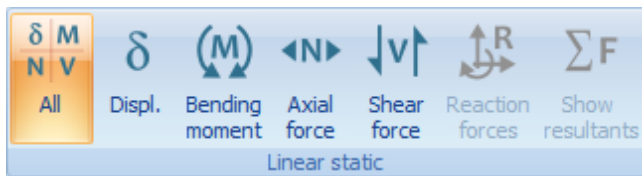
#### Metadata

ID:	18
Klassifisering:	Feil
Sist modifisert:	20. mars
Modifisert av:	Torjus
Rapportert i versjon:	3.0
Rapportert dato:	Vår 2013
Rapportert av:	Erik Aasmundrud

#### Endrede klasser

```
fap2D.Gui FrmMainGui.designer
          FrmMainGui.ResultsPage
```

## Figurer



Figur A.13: **3.0:** *Show resultants* og *reaction forces* vises som deaktiverte.



Figur A.14: **3.1:** *Resultants* og *reaction forces* i egen gruppe, som bare vises når det er nødvendig.

### A.1.7 Zooming over *selection* i *toolbox panel*

#### Endring

Disse gruppene er byttet om i *toolbox panel* slik at zooming står over *selection*

#### Argumentasjon

Grunnen til endringen er at zooming brukes både i *modelling display* og *results display*, men det gjør ikke *selection*. Derfor bør zooming stå øverst slik at den ikke endrer posisjon når det veksles mellom de to *displayene*. Skaper bedre flyt i programmet.

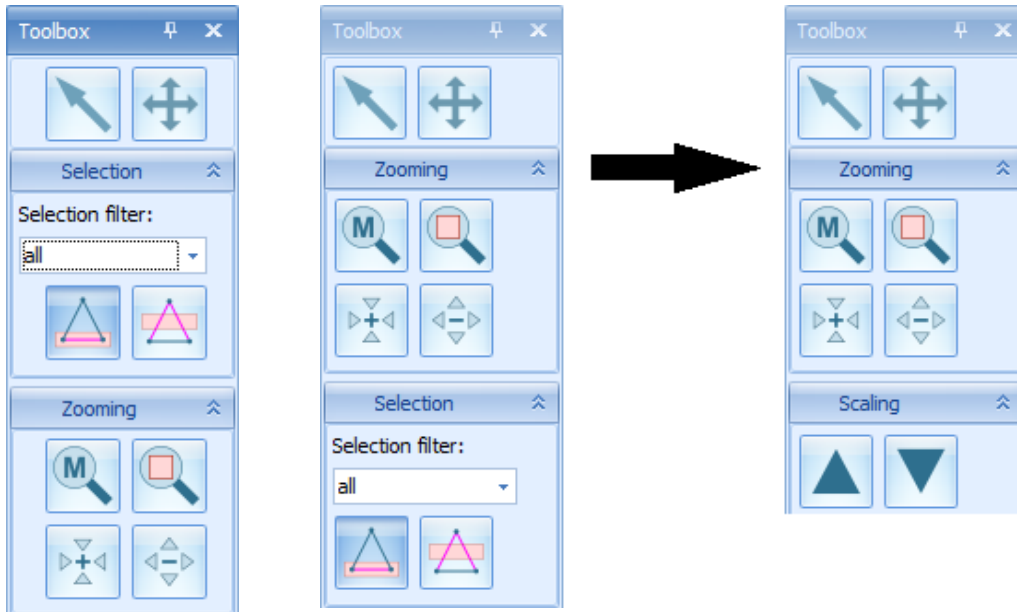
#### Metadata

ID:	77
Klassifisering:	Endring
Sist modifisert:	7. mai
Modifisert av:	Torjus
Rapportert i versjon:	3.0
Rapportert dato:	Vår 2013
Rapportert av:	Torjus

#### Endrede klasser

fap2D.Gui FrmMainGui.Designer

## Figurer



Figur A.15: **3.0:**  
Selection er over  
zooming.

Figur A.16: **3.1:** Zooming over selection i både  
modellering (t.v) og results display (t.h).

### A.1.8 Bruk av *add-* og *edit-*symbol i *line load dialog*

#### Feil

*Line load dialog* brukes både når man legger til nye *line loads* (via *line load button* i *loading ribbon page*), og når man endrer *line loads* (ved å høyreklikke på en *line load* i modellen).

Når man åpner dialogen for å endre en *line load* bekrefter man endringene ved å trykke på en "OK"-knapp. Denne knappen har alltid et plusstegn og det er misvisende fordi plusstegnet skal indikere at det legges til en ny *line load*.

#### Endring

Når brukeren åpner dialogen for å legge til ny last vises fortsatt plusstegnet. Når brukeren derimot åpner den for å endre en last vises "hake"-symbolet.

#### Figurer

#### Argumentasjon

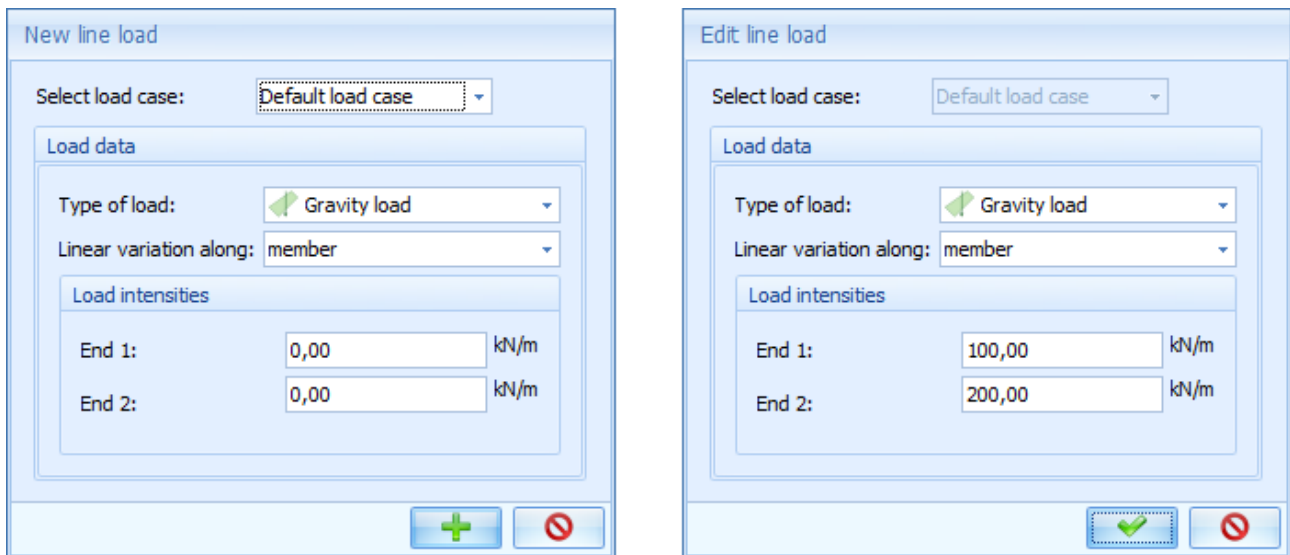
Feilen er kanskje vanskelig å få øye på, men er rettet fordi det var enkelt og det gjør GUI mer korrekt og fjerner all grunn til tvil hos brukeren.

#### Metadata

ID:	100
Klassifisering:	Feil
Sist modifisert:	14. mai
Modifisert av:	Torjus
Rapportert i versjon:	3.0
Rapportert dato:	Vår 2014
Rapportert av:	Kolbein

#### Endrede klasser

```
fap2D.Gui FrmLineLoadDialog,
.resx
```



Figur A.17: **3.1:** Ny *line load* (t.v) - plusstegn i OK-knappen. Endring av *line load* - hake-symbol i OK-knappen.

### A.1.9 Nytt symbol for *settings button*

#### Endring

Det er lagt til et nytt symbol for *settings button* (som brukes i *quick access toolbar* og *application menu*). Symbolet er et tannhjul.

#### Argumentasjon

Det nye symbolet er mye mer visuelt og gjenkjennelig for de fleste. Det er et vanlig symbol å bruke for en *settings/options button*).

#### Metadata

ID:	116
Klassifisering:	Endring, Nytt
Sist modifisert:	7. mai
Modifisert av:	Espen
Rapportert i versjon:	3.0
Rapportert dato:	Mai 2014?
Rapportert av:	Espen

#### Endrede klasser

```
fap2D.Gui FrmMainGui,
        .Designer, .resx
```

#### Figurer



Figur A.18: **3.1:** Nytt symbol for *settings button*. Det gamle symbolet var nesten utydelig og lite intuitivt.

### A.1.10 Oppdaterer y-aksen i graf for ikke-lineær fjær

#### Feil

Y-aksen for ikke-lineær fjær ble ikke oppdatert til riktig tekst etter at typen fjærstivhet ble endret fra rotasjonsstivhet til aksialstivhet. Derimot ble enheter endret.

#### Endring

Feilen var begrenset til teksten, og hadde ingen påvirkning på de faktiske verdiene. Det var likevel prioritert å fjerne inkonsistens for å unngå forvirring.

Endring av typen fjærstivhet bytter enhetene og teksten i grafen, og tvinger den til å oppdateres.

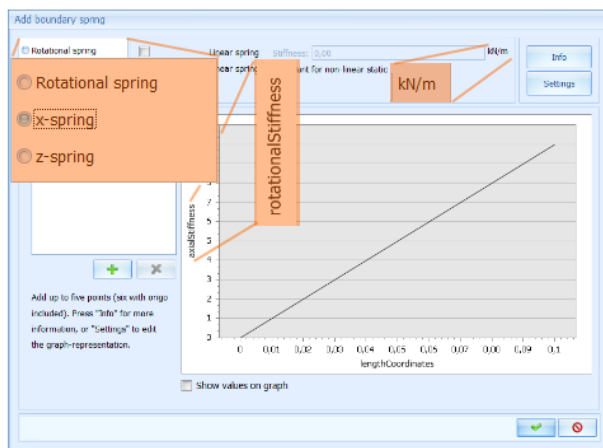
#### Metadata

ID:	126
Klassifisering:	Feil
Sist modifisert:	19.05
Modifisert av:	Espen
Rapportert i versjon:	3.0
Rapportert dato:	11.05
Rapportert av:	Espen

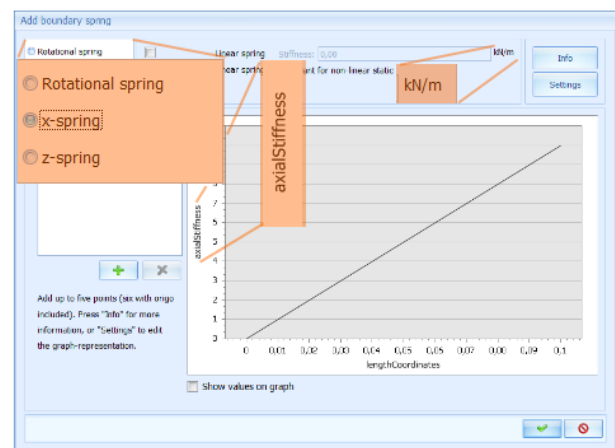
#### Endrede klasser

fap2D.Gui UC\_NonLinearElasticSpring.cs

#### Figurer



Figur A.19: Før - Tekst på y-akse er ikke oppdatert.



Figur A.20: Etter - Y-aksen har korrekt tekst.

### A.1.11 Gå til resultatvisning etter analyse, og zoom to model

ID	20 (+ 101)
Klassifisering	Endring
Modifisert dato	31.01
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	Januar 2014
Rapportert av	Kolbein Bell

#### Endringer - Versjon 3.3.0 Developer

Etter at en analyse er ferdig kjørt vil resultatvinduet automatisk vises frem. Før måtte man manuelt velge *Results*-fanen for å få vist resultatvinduet i stedet for modellvinduet.



Feilen behandles vha. den nye metoden `PrepareResultsPage(...)`. Denne metoden er også brukt for å rette feil A.1.12.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmMainGui.AnalysisPage

### Detaljer - Versjon 3.3.0 Developer

Endringen ble gjort fordi man alltid er interessert i å se resultater etter en analyse. Det er ingen grunn til at brukeren skal måtte trykke på *results ribbon page* for å få vist dem. Skaper bedre flyt i programmet.

### Endringer - Versjon 3.1

Når man kjører analyse og blir sendt til resultatvisningen så blir funksjonen *Zoom to model* kjørt slik at resultatene blir vist for hele modellen.

Metodekallet til *Zoom to model* foretas nå i metoden `PrepareResultsPage(...)`.

Det er også lagt til en innstilling der brukeren kan velge å ikke endre zoom-nivået før resultatene vises. Se XXX ref til *setting* her.

<i>Namespace</i>	Endrede klasser og filer
fap2D.Gui	FrmMainGui.AnalysisPage FrmMainGui.ResultsPage FrmSettings, .Designer, .resx
Resources	global.fap
fap2D.Utilities	GlobalSettings GlobalSettings.ComputationalModel
fap2D.Storing	GlobalSettingsDataSet.Designer, .xsd, .xss StoringUtilities

### Detaljer - Versjon 3.1

Endringen er gjort for å gjøre resultatvisningen mer tilpasset til individuelle behov. Av og til kan det være praktisk å vise hele modellen, mens andre ganger er man mer interessert i å vise resultater for et utsnitt. Med den nye innstillingen kan man variere dette.

## A.1.12 Samsvar mellom knapp og diagram i resultatvisninger

ID	19 (+41)
Klassifisering	Feil
Modifisert dato	04.02
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	Vår 2013
Rapportert av	Erik Aasmundrud

## Feil

Visningen av resultatdiagrammer i *results ribbon page* fungerer ikke optimalt når man veksler fra *results* til en annen *ribbon page* og så tilbake til *results* igjen:

Knappen som tilhører *All*-diagrammet vises som aktivert uansett hvilket resultatdiagram som ble vist i *results ribbon page* før vekslingen. Denne feilen gjelder alle analyser som viser resultatdiagrammene *All*, *displacement*, *bending moment*, *axial force* og *shear force*.

Eksempel:

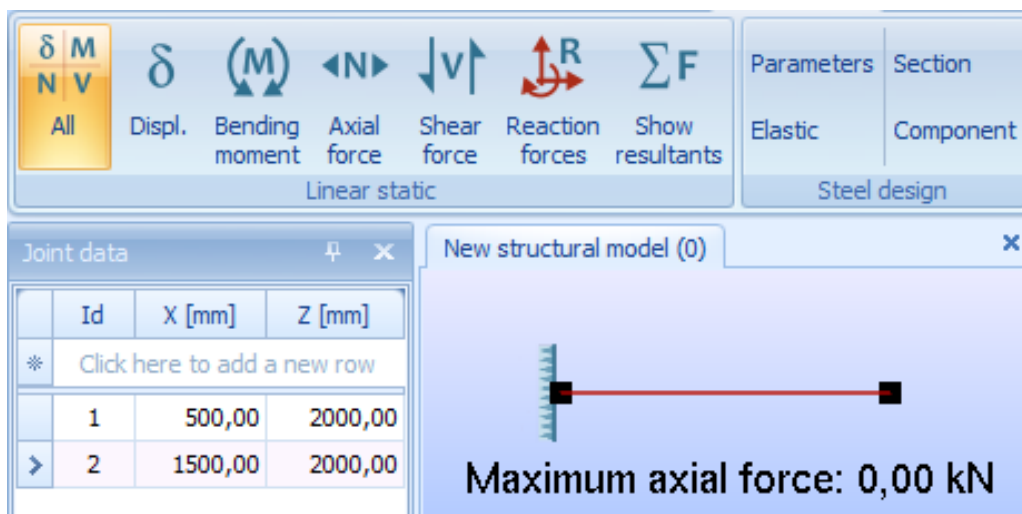
1. Utfør en lineær statisk analyse. Resultatdiagrammet og knappen *All* vil vises som aktivt.
2. Bytt til et annet diagram, for eksempel *Axial force*. Knappen for *Axial force* aktiveres.
3. Veksle til *modelling ribbon page* og så tilbake til *results*.
4. *Axial force*-diagrammet vil fortsatt vises, men *All*-knappen er aktivert i stedet for *Axial force*-knappen.

## Endringer

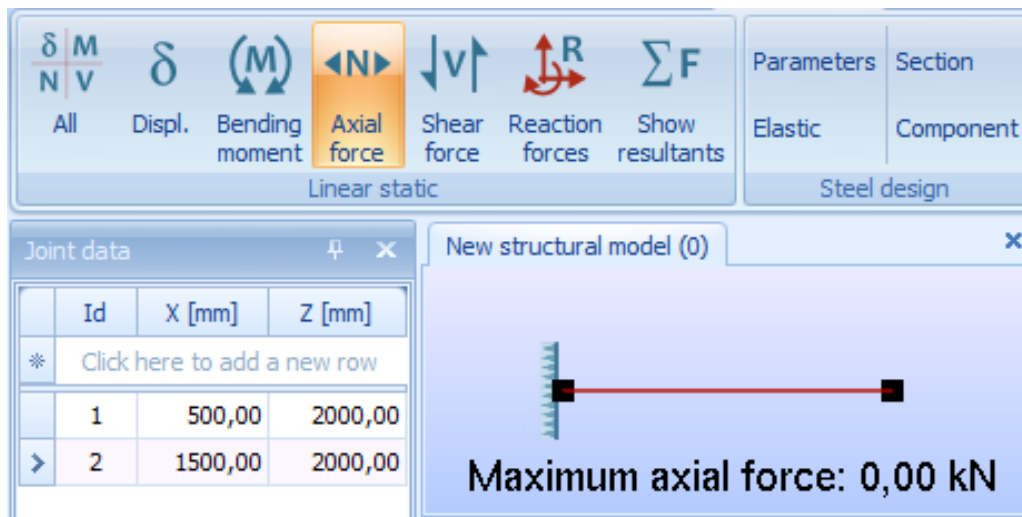
Det aktive resultatdiagrammet vises sammen med sin tilhørende knapp, både etter veksling mellom *ribbons* og bruk av *Neutral pointer*.

Den nye metoden `PrepareResultsPage(...)` er brukt for å rette feilen og for å klargjøre resultatvisningen generelt etter analyse. Se feil A.1.11 for mer informasjon. Metoden sørger for gjenbruk av kode som er felles for flere analysetyper, slik at det er enklere å vedlikeholde koden.

Namespace	Endrede klasser
fap2D.Gui	FrmMainGui.AnalysisPage FrmMainGui.ResultsPage



Figur A.21: Før endring: Etter veksling mellom *ribbon pages* eller bruk av *Neutral pointer*-knappen er det ikke samsvar mellom knapp (*All*) og diagram (*Axial force*).



Figur A.22: Etter endring: Samsvar mellom knapp og diagram (*Axial force*).

## Detaljer

Feilen er av kosmetisk art, men er viktig å rette fordi den er såpass enkel å rette opp og fordi den gir brukerne mer tillit til programmet.

### A.1.13 Lukking av dialogboks for initialtøyning og -temperatur

ID	30
Klassifisering	Feil
Modifisert dato	20.03
Modifisert av	Espen, Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	Mars 2014
Rapportert av	Espen

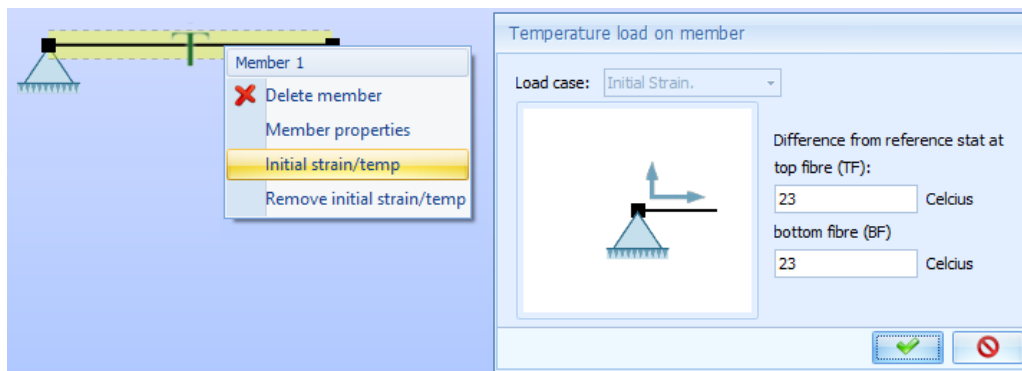
## Feil

Dialogboksen lukkes vha. metoden `Dispose()` når man trykker på avbryt-knappen. Metoden brukes for å frigjøre ressurser som dialogboks-instansen forbruker, men er med på å gjøre brukergrensesnittet tregere (hakkende bilde).

## Endringer

Dialogboksen lukkes vha. metoden `Close()` når man trykker på avbryt-knappen. Metoden er enklere enn `Dispose()` og passer bedre for å brukes på den tråden som brukergrensesnittet benytter seg av (bildet blir glattere). Endringen er i tråd med praksisen i de andre dialogboksene i **fap2D**.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmTempStrainIndividualMembers



Figur A.23: Dialogboksen for initialtøyning og -temperatur.

## Detaljer

Feilen er av kosmetisk art, men er rettet fordi det var fort gjort. Rettingen skaper en bedre brukeropplevelse. Ressursene som metoden `Dispose()` frigjør er små i forhold til `Close()` så det påvirker ikke minnebruken til **fap2D**.

### A.1.14 Flere åpne modeller samtidig

ID	60
Klassifisering	Feil
Modifisert dato	19.05
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	21.04
Rapportert av	Petter

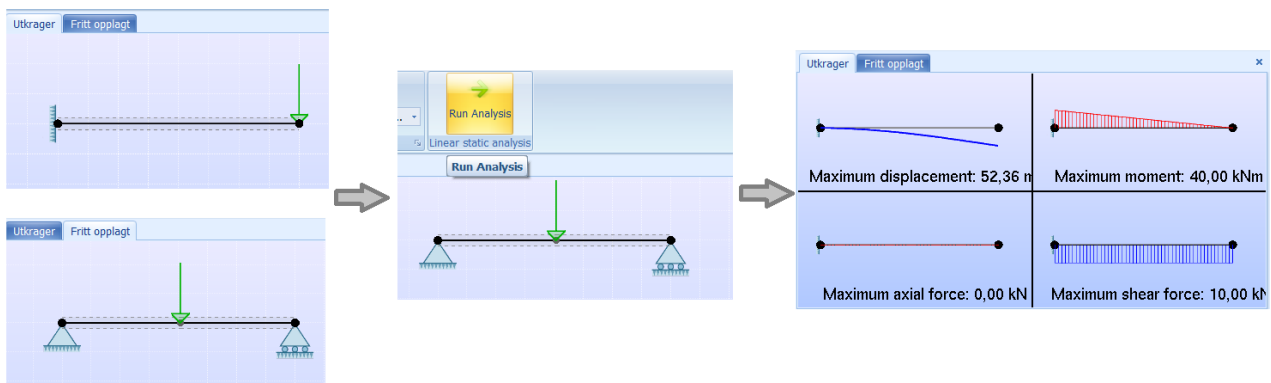
## Feil

Analyse ble ofte utført på feil modell når flere av dem var åpne. Vanligvis ble kun den første kjørt.

## Endringer

Under analyse blir alle *controls* og paneler skrudd av for å unngå at bruker påvirker analysen. En kodelinje som skrudde av et panel byttet fanefokus, og dermed også modell. Den linjen ble fjernet.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmMainGui.cs FrmMainGui.AnalysisPage.cs



Figur A.24: Hendelsesforløp – programmet regner på modellen i fane 1.

## Detaljer

### Kun første modell kjøres

*Debugging* viser at når *toolbox* skrus av byttes modellfanen til modellen som først ble åpnet.

```
...
//dockPanelModellingToolbox.Enabled = false;
...
```

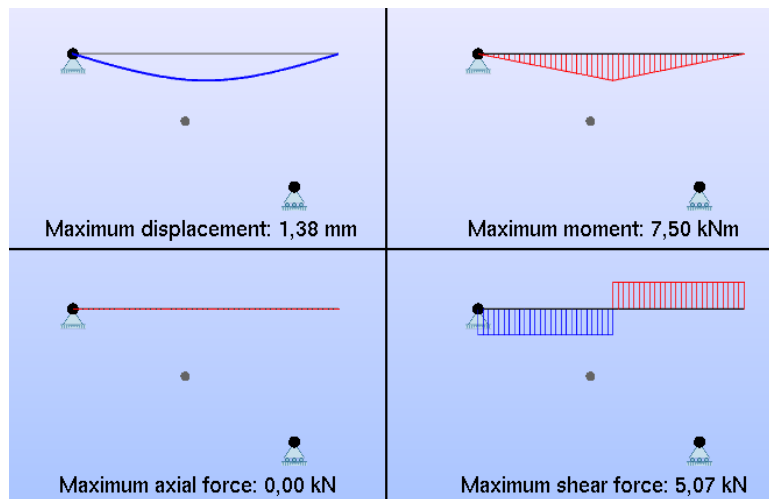
Hvorfor dette skjer er uvisst, men er trolig at det er en subtil feil forbundet med rekkefølgen på *panels* men det er tidkrevende å sette seg inn i. Løsning blir å ikke deaktivere *toolboksen*, men heller dens komponenter (knapper, tekstbokser, etc.) Dette regnes som tilnærmet ekvivalent oppførsel og løser problemet med hvilken modell som kjøres.

### Gamle resultater vises

Å nulle ut resultatene for konstruksjonsmodeller som har blitt endret løser problemet der gamle, ugyldige resultater vises.

```
public void UpdateUndoRedoBuffer()
...
ComputationalModel = null;
...
```

Feilen var at resultatfanen for en endret konstruksjonsmodell med gamle resultater kunne re-aktiveres ved å gå via resultatfanen til en annen modell. Programmet trodde at brukeren skulle sammenligne resultater, noe som er en nyttig funksjonalitet i **fap2D**, men resultatene som ble vist var utdaterte. Løsningen var å slette beregningsmodellen når endringer ble gjort i konstruksjonsmodellen.



Figur A.25: Utdaterte resultater vises for en fritt opplagt bjelke der *joint 2* er flyttet ned.

### A.1.15 Unødvendig reposisjonering av *pring dialog*

#### Feil

*Spring dialog* endrer størrelse når man trykker på noen av knappene i dialogen. Hver gang den endrer størrelse blir den reposisjonert til midten av skjermen.

#### Endring

Metoden `IsWithinScreenBounds(...)` er blitt laget. Den brukes for å sjekke om en dialogboks ligger helt innenfor skjermen eller ei. Dersom dialogen er innenfor blir den ikke sentrert på nytt.

#### Argumentasjon

Reposisjoneringen skaper en hakkete brukeropplevelse. Endringen bedrer dette. Metoden som er laget kan også brukes for andre dialogbokser, men det er ikke så mange som har behov for dne per nå.

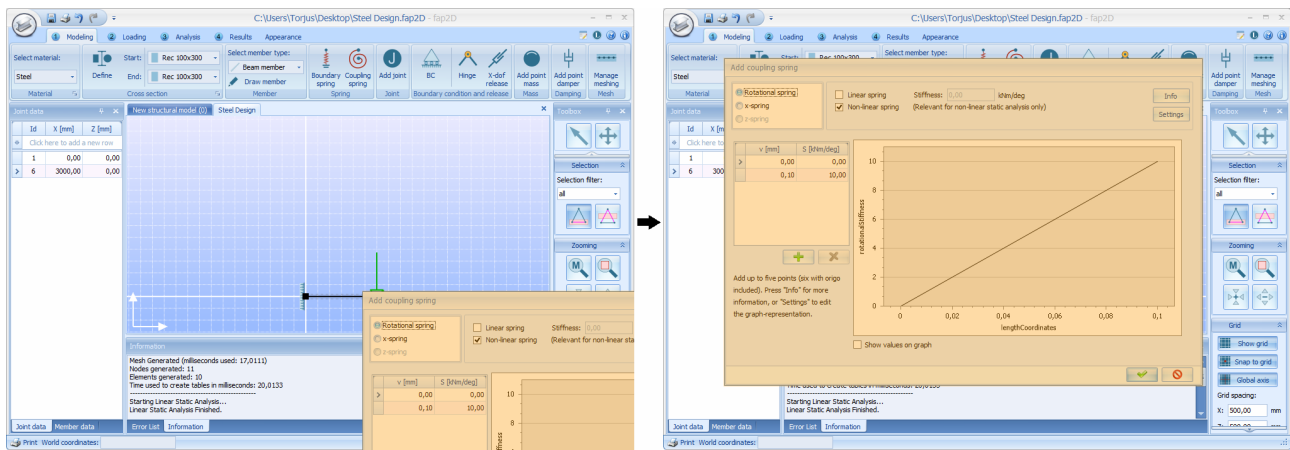
#### Figurer

#### Metadata

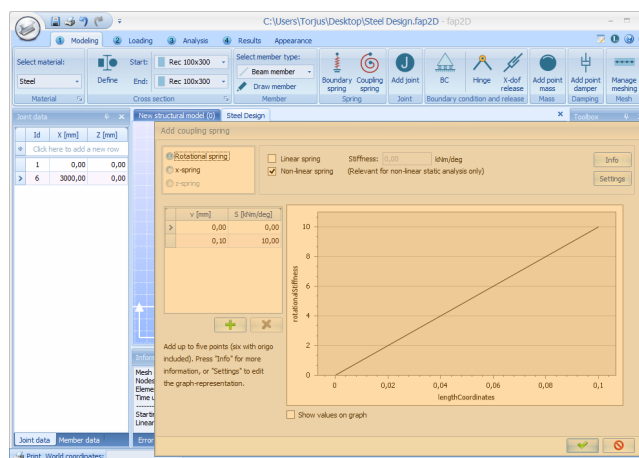
ID:	72
Klassifisering:	Feil
Sist modifisert:	3. april
Modifisert av:	Torjus
Rapportert i versjon:	3.0
Rapportert dato:	2. april
Rapportert av:	Torjus

#### Endrede klasser

<code>fap2D.Gui</code>	Utilities
<code>fap2D.Utilities</code>	<code>FrmSpringDialog</code>



Figur A.26: **3.0** og **3.1**: Hele dialogen er ikke innenfor skjermen (t.v) og blir derfor sentrert (t.h).



Figur A.27: Bare i **3.1**: Hele dialogen vises på skjermen, ikke behov for sentrering.

### A.1.16 *Reselecting point load* og member buttons

#### Feil

Etter at man har trykket på *point load button* eller *draw member button* og plassert ut ett av de to objektene for 1. gang må man trykke på knappen to ganger for å få lagt til et nytt objekt. Man må altså deaktivere knappen før den kan brukes på nytt.

#### Endring

Etter endring trenger man bare å trykke én gang når man skal legge til en ekstra *member* eller *point load*.

#### Argumentasjon

Feilen skaper dårlig flyt fordi man må trykke flere ganger enn nødvendig. Kan irritere brukeren.

Det har nok vært meningen at knappene blir vist som aktive (merket i gult) mens man setter ut *member* eller *point load*, men det er overflødig fordi det vises uansett en tekst eller et symbol ved musepekeren for å indikere hva som er den aktive knappen.

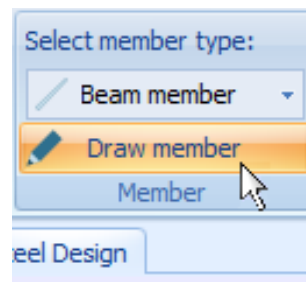
#### Metadata

ID:	89
Klassifisering:	Feil
Sist modifisert:	9. mai
Modifisert av:	Torjus
Rapportert i versjon:	3.0
Rapportert dato:	Vår 2014
Rapportert av:	Kolbein

#### Endrede klasser

```
fap2D.Gui FrmMainGui.LoadingPage
FrmMainGui.ModellingPage
```

#### Figurer



Figur A.28: **3.0:** *Member button* blir markert i gult når man trykker på den. Må deaktiveres før den kan brukes igjen.



### A.1.17 Konflikt mellom *joint/member buttons* og *boundary conditions button*

#### Feil

Når man trykker på knappen som viser frem *boundary conditions*-panelet, etter å ha trykket på enten *joint button* eller *member button* vil førstnevnte knapper holde seg aktive selv om de burde blitt deaktivert.

Feilen er en følgefeil pga. feilen i punkt A.1.16.

#### Endring

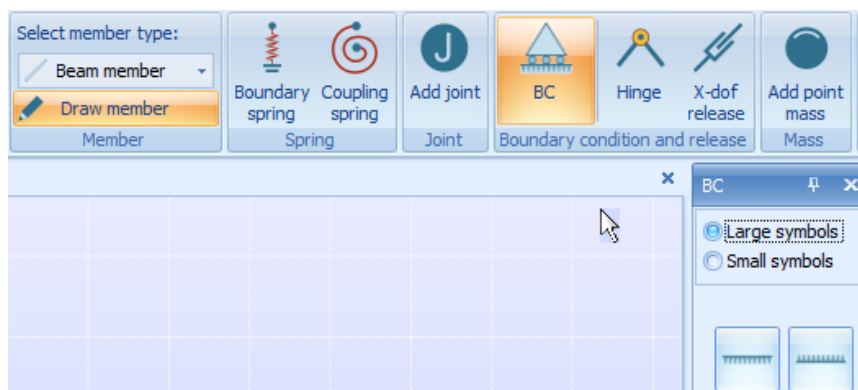
GUI blir nå *reset* slik at *joint* og *member buttons* blir deaktiverte før man trykker på en ny knapp.

#### Argumentasjon

Feilen har egentlig ingen ting å si for den hindrer ikke brukeren i å få satt på en *boundary condition*. Man må nemlig velge en spesifikk *boundary condition* fra panelet for å sette den på, og når man gjør dette vil *joint/member button* deaktiveres.

Likevel er feilen rettet fordi det gjør GUI ryddigere.

#### Figurer



Figur A.29: 3.0: Både *member* og *boundary conditions button* vises som aktive og det kan forvirre brukeren.

#### Metadata

ID:	92
Klassifisering:	Feil
Sist modifisert:	12. mai
Modifisert av:	Torjus
Rapportert i versjon:	3.0
Rapportert dato:	Vår 2013?
Rapportert av:	

#### Endrede klasser

fap2D.Gui FrmMainGui.ModellingPage

## A.1.18 *Unhandled exception for remove button* i *earthquake loading dialog*

### Feil

Det oppstår en *unhandled exception* når man trykker på *remove excitations button* i *earthquake loading dialog*, når det ikke er noen *excitations* (jordskjelv) å fjerne. Grunnen til feilen er at man prøver å slette en verdi som ikke eksisterer i kildekoden - man peker på et dataobjekt som er null.

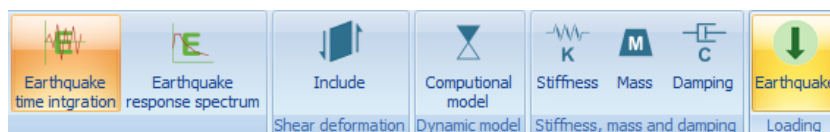
### Endring

Det er innført logikk som sørger for at sletting ikke er mulig når ingen *excitations* er lagt til i modellen. Dessuten er det sørget for at *remove button* bare er aktiv når det er blitt lagt til *excitations*.

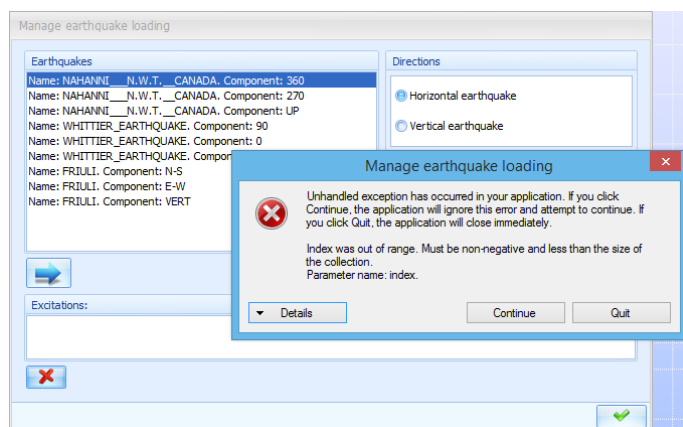
### Argumentasjon

Rettet fordi den ødelegger programflyten ved å vise en varselboks. Programkjøringen kan heldigvis fortsettes ved å trykke OK i boksen, så feilen er ikke alvorlig.

### Figurer



Figur A.30: 3.0 og 3.1: Knappen som åpner *earthquake loading dialog* (i gult).



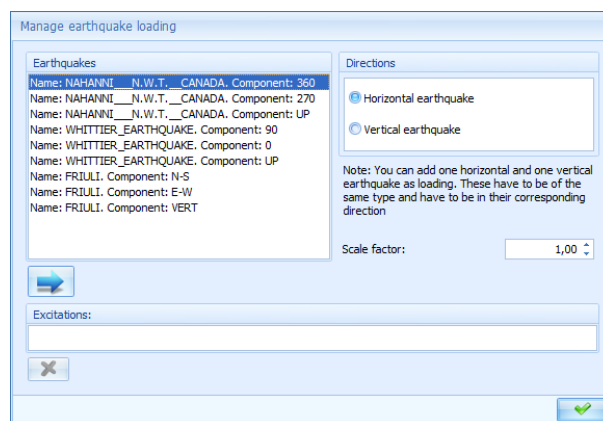
Figur A.31: 3.0: Feilmelding ved trykking på *remove button* fordi ingen laster er lagt til i listen nederst.

### Metadata

ID:	96
Klassifisering:	Feil
Sist modifisert:	14.05
Modifisert av:	Torjus
Rapportert i versjon:	3.0
Rapportert dato:	Vår 2014
Rapportert av:	Torjus?

### Endrede klasser

fap2D.Gui FrmLoadTimeEarthquake, .Designer, .resx



Figur A.32: 3.1: *Remove button* deaktivert, siden ingen *excitations* er lagt til i listen nederst (dialogen er krympet litt).

### A.1.19 *Zoom to model* ved visning av resultater

#### Endring

Når en analyse er ferdig og resultater skal vises i *results display* blir nå modellen zoomet til et nivå som gjør at hele modellen vises. Dersom man veksler tilbake til *modelling display* etterpå vil det nye zoom-nivået fortsatt gjelde slik at hele modellen vises. Den nye logikken er enkelt lagt til ved å kalle en metode som brukes fra før (i en knapp i *toolbox panel*). Kallet er lagt til i metoden `PrepareResultsPage(...)` som brukes til å utføre operasjoner som til sammen klargjør *results ribbon page* etter en analyse.

Det er lagt til en innstilling i *settings* der man kan deaktivere *zoom to model* ved visning av resultater, som f.eks. er praktisk dersom man bare er interessert i å jobbe med et utsnitt av modellen. Se punkt A.9.5 for detaljer rundt implementasjon av denne innstillingen i *settings*.

#### Argumentasjon

*Zoom to model* vil sørge for at resultatene presenteres på en pen og oversiktlig måte. *Zoom to model* passer godt i de fleste tilfeller, og dersom den ikke gjør det kan man bare deaktivere den i *settings* (og åpne den samme praksisen som i versjon **3.0**). Mer fleksibelt for brukeren.

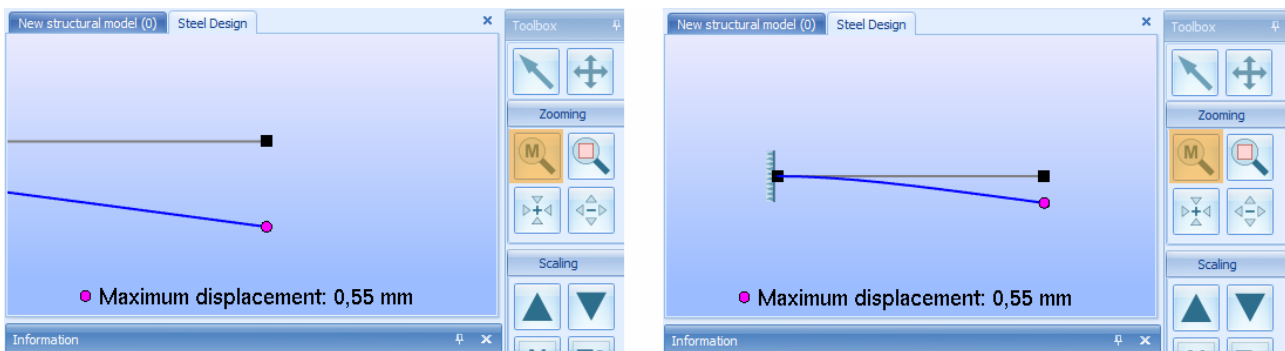
#### Metadata

ID:	101
Klassifisering:	Endring
Sist modifisert:	14. mai
Modifisert av:	Torjus
Rapportert i versjon:	3.0
Rapportert dato:	Vår 2014
Rapportert av:	Kolbein

#### Endrede klasser

fap2D.Gui FrmMainGui.AnalysisPage

#### Figurer



Figur A.33: **3.1:** *Zoom to model* er deaktivert (t.v) og zoom-nivået er vilkårlig etter analyse. T.h: *Zoom to model* er aktivert og hele modellen vises. Knappen i *toolbox* (i oransje) er den som primært brukes for å oppnå at hele modellen vises.

## A.1.20 Endring av output-verdier i *resultants dialog*

### Feil

Det er mulig å endre output-verdiene i *resultants dialog*. Det gjør at brukeren kan komme i skade for å endre verdiene.

### Endring

Det er ikke lenger mulig å endre verdiene, men det er mulig å kopiere. Dette oppnådd ved bruk av `ReadOnly`-egenskapen til tekstfeltene (`TextEdits`-objekter) som inneholder verdiene.

### Argumentasjon

Endring av output-verdier skal ikke forekomme, det ligger i definisjonen av ordet `output`!

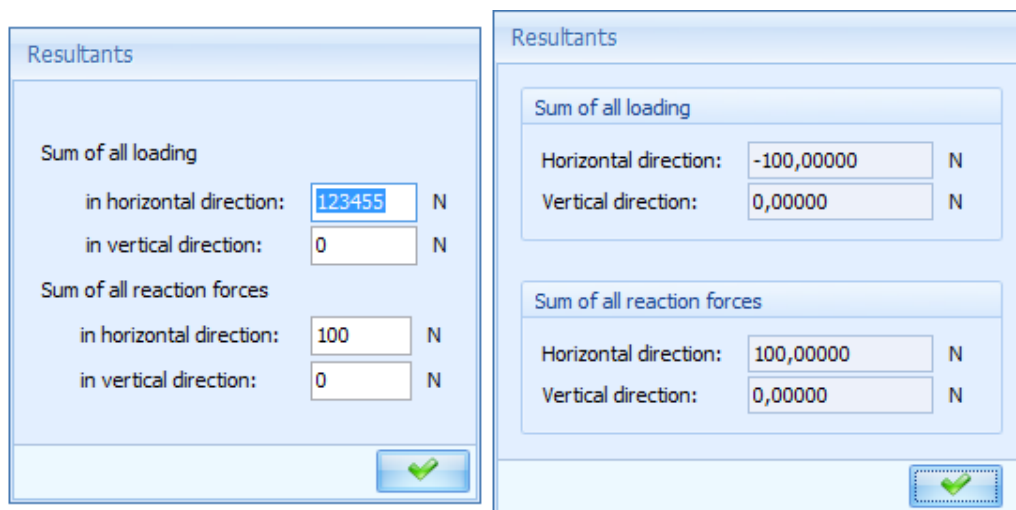
### Metadata

ID:	23
Klassifisering:	Feil
Sist modifisert:	4. april
Modifisert av:	Torjus
Rapportert i versjon:	3.0
Rapportert dato:	Vår 2013
Rapportert av:	Espen?

### Endrede klasser

<code>fap2D.Utilities</code>	<code>Utilities</code>
<code>fap2D.Gui</code>	<code>FrmSpringDialog</code>

### Figurer



Figur A.34: **3.0** (t.v): Mulig å endre output-verdi. **3.1** (t.h): Bare mulig å kopiere output-verdi.

## A.1.21 DevExpress-oppdateringer

ID	63
Klassifisering	Endring
Modifisert dato	24.03
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	07.01
Rapportert av	Kolbein/DevExpress

## Endringer

**fap2D** benytter seg av rammeverket DevExpress, og derfor har kildekoden blitt oppdatert for å støtte nye versjoner av rammeverket.

Versjon	Dato innført
13.1.8	8. nov. 2013.
13.2.5	23. des. 2013.
13.2.6	16. jan.
13.2.7	24. mars
13.2.8	24. mars

Tabell A.4: Oppdatering av DevExpress.

Den eneste versjonen som førte til at kildekoden måtte endres var versjon **13.2.6**. Konverteringsverktøyet som DevExpress har laget fungerte ikke godt for denne oppdateringen, så vi ble nødt til å endre alle referanser til DevExpress i kildekoden fra **13.2.5** til **13.2.6**, manuelt. Dessuten måtte man ta hensyn til endringer i rammeverket:

- `Axis.Range.ScrollingRange` måtte erstattes med `Axis.WholeRange`.
- `Axis.Range` måtte erstattes med `Axis.VisualRange`.
- `XtraGrid.Views.Grid.GridView().ShowButtonMode` måtte erstattes med `XtraGrid.Views.Grid.GridView().OptionsView.ShowButtonMode`.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmResParaLinRSpectrumEarthquake.Designer FrmGraphDialog.Designer FrmInfluenceLineLoadTrainDialog.Designer FrmRealTimeLoadTrainDialog.Designer
fap2D.ModelData	UC_Graph.Designer UC_GraphNonLinearStiffness.Designer

## A.2 Tegning - OpenGL

### A.2.1 Forbedring av N- og V-diagrammer

ID	1
Klassifisering	Feil
Modifisert dato	27.01
Modifisert av	Espen, (Torjus)
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	August 2013
Rapportert av	Kolbein Bell

## Feil

Skjær- og aksialkraft-diagrammene i resultatvinduet ble ved noen tilfeller tegnet på feil side av elementene. Diagrammet kunne bytte side langs *arch members*, eller mellom *members*. Resultatet var forvirrende diagrammer.

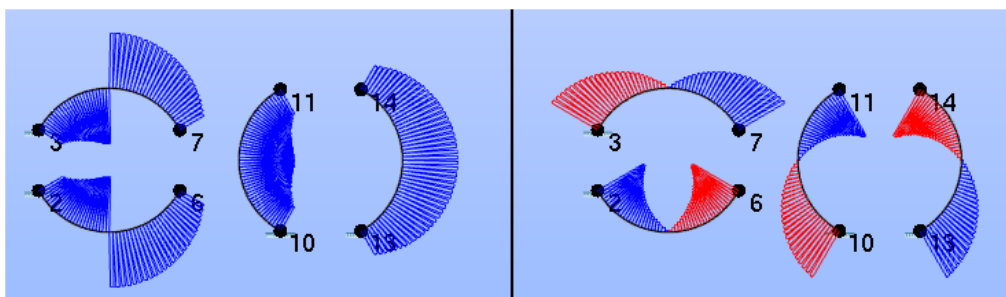
## Endringer

For å hindre at diagrammene skiftet side langs en *arch member* er det valgt å tegne diagrammet på samme måte som bøyemoment. Da tegnes positive verdier på oversiden, og negative verdier på undersiden.

Diagrammer skiftet retning på tvers av *members* fordi de kun ble tegnet avhengig av vinklene til deres elementer. Siden denne vinkelen kunne være alt fra  $0^\circ$  til pluss/minus  $180^\circ$  kunne diagrammene til to elementer med nesten lik vinkel bli tegnet på ulik måte. Det er endret slik at elementene defineres med én vinkel, dvs. alle elementer har vinkel innenfor en halvsirkel. *Grensevinklene*<sup>1</sup> til denne halvsirkelen er også endret til  $-45^\circ$  og  $135^\circ$  for at *members* med tilnærmet vertikal eller horisontal vinkel tegnes konsekvent.

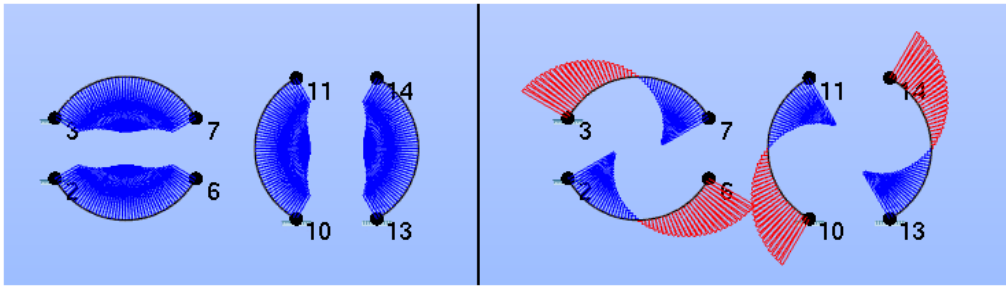
De nye grensevinklene kan ikke garantere at diagrammer bytter retning på tvers av dem, (se figur A.38). Derfor er det implementert en klasse som skal brukes i et enkelt søketre, `MemberTreeNode.cs`. Ideen er at de *members* som er koblet sammen, spesielt ved grensevinklene, skal tegne sine N- og V-diagrammer i samme retning. En eller flere "*rotmembers*" blir valgt til å bestemme hvilken retning sine "*bladmembere*" skal tegne diagrammet i.

<i>Namespace</i>	Endrede klasser
fap2D.Computations	Element
	ComputationalModel
	ComputationalModel.Drawing
	ComputationalModel.ModelTools
	MemberTreeNode

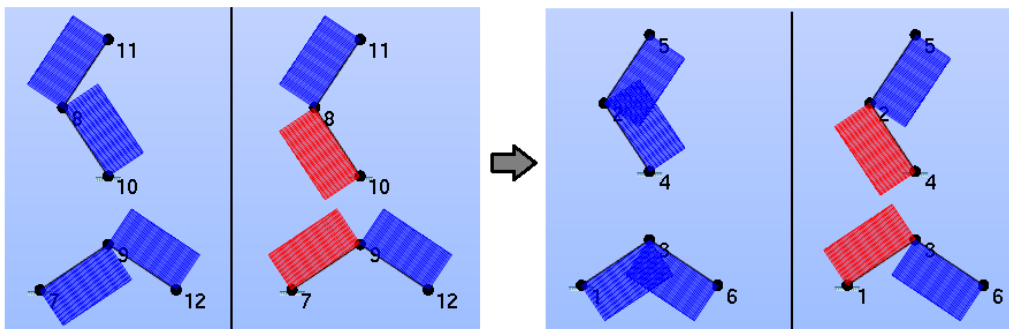


Figur A.35: Før endring - hhv. N- og V-diagram skifter side langs *arch member* ved  $0^\circ$  eller  $180^\circ$ .

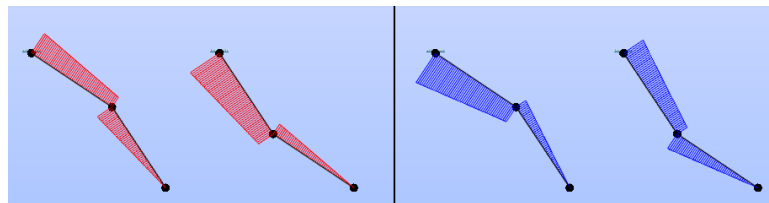
<sup>1</sup>Vinklene som elementer kan ha retning innenfor



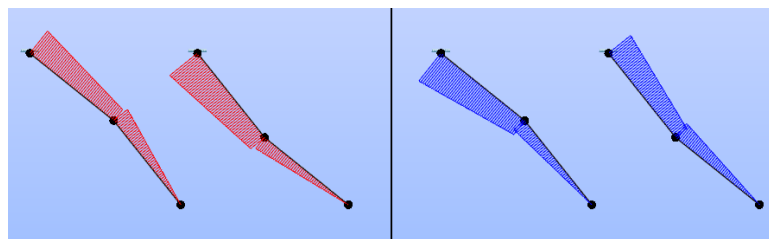
Figur A.36: Etter endring - hhv. N- og V-diagram tegnes som for momenter, positivt på oversiden, negativt på undersiden.



Figur A.37: t.v. Før endring - hhv. N- og V-diagram skifter side ved  $0^\circ$  eller  $180^\circ$ . - t.h. Etter endring - hhv. N- og V-diagram skifter side ved  $-45^\circ$  eller  $135^\circ$ .



Figur A.38: Før endring - hhv. N- og V-diagram skifter side ved *grensevinklene*.



Figur A.39: Etter endring - hhv. N- og V-diagram er konsistent på tvers av *members*.

## Detaljer

N- og V-diagrammet fulgte ingen retningslinjer for hvordan de skulle tegnes. Dermed kunne diagrammene tilsynelatende tegnes på *feil* side, avhengig av hvordan *members* modelleres, og hvilken retning de hadde. Diagrammene kunne forvirre brukeren.

**Merk:** Diagrammer tegnes for hvert element, og er kun løst tilknyttet *members*

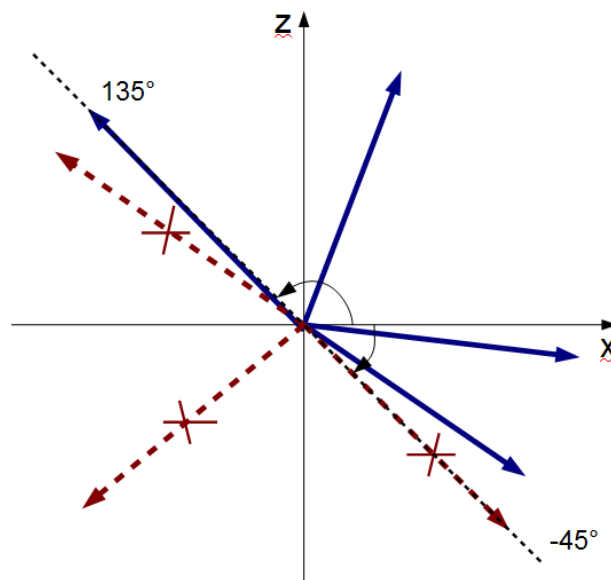
**Del 1 – Arch members:** Elementer langs en *arch member* har ulik vinkel langs *memberen*. Siden diagrammer tegnes med utgangspunkt i vinklene kan diagrammene bytte side internt i *arch membere*. Hvis man tegner N- og V-diagrammene på samme måte som M-diagrammer får man konsistente diagrammer der positive krefter alltid tegnes på oversiden, og negative på undersiden (i forhold til *memberens* lokale aksesystem).

Endringen vi gjorde i tegnemethoden til Element.cs gjorde at elementer kunne ha alle vinkler fra  $0^\circ$  til  $360^\circ$ . Dette måtte vi endre tilbake.

**Merk:** En senere bestemmelse om konsekvent retning av *members* gjør at alle *members* har retning mellom  $-90^\circ$  og  $+90^\circ$ . Se endring 85.

**Del 2 – Members langs globale akser:** Brukere har forventninger til diagrammer for elementer som er opprettet av *members* med retning nær de globale aksene. Alle *members* med omtrent samme vinkel som x-aksen forventes å tegnes med diagrammene på samme side, og tilsvarende for vertikale *members*. For å implementere dette tegnes alle elementer med en vinkel innenfor et område  $-45^\circ$  og  $135^\circ$ .

Hva som er intuitiv forståelse kan variere avhengig av modellen, men de nevnte reglene dekker tilfellene der bruker enkelt vil oppdage uoverensstemmelser.



Figur A.40: Retning som elementer tegnes i.

**Del 3 – "Grense-members":** For å ta hensyn til de tilfellene der *members* ligger langs grensevinklene er det bestemt å bruke ett eller flere søketrær for å finne *members*, og tilhørende elementer, som henger sammen og skal ha like diagrammer på tvers av grensene.

Det er lagt til en ny klasse `MemberTreeNode.cs` i `fap2D.Computations` for å bygge trestrukturen. Hver *member* er en node i treet, og de er koblet sammen via *joints*. Node i denne sammenhengen er som søketre-node, ikke elementnode. Kun *members* langs grensevinklene ( $-45^\circ$  og  $135^\circ$ ), pluss/minus en buffer satt til  $20^\circ$ , brukes i søketrærne fordi resten av *members* vil oppføre seg pent.

Rotnodene i søketrærne er en eller flere av de utvalgte *members* som ikke er knyttet til en annen rotnode. *Memberen* med lavest *member-ID* som ikke er i et tre blir valgt som rot. Hvert tre bygges ved *depth-first, pre-order*, dvs. først en rot, så venstre subtre, og deretter høyre subtre.



Metoden `GetDirectionChangingMembers(...)` i `MemberTreeNode`-objekter brukes av *rotmemberne* til å returnere *ID'ene* til *members* som bør bytte retning på diagrammene.

Prosessen med å bygge opp og finne *members* foregår i metoden `GenerateDiagram-ChangingElements(...)` i `ComputationalModel.ModelTools.cs`. Den kalles etter *meshing* for å generere en liste med *element-ID'er* (`_diagramChangingElementId`) som korresponderer med *memberne* returnert av kallene til `GetDirectionChangingMembers(...)`.

Metoden som tegner diagrammer endres slik at hvert element med ID i listen `_diagramChangingElementId` skaleres slik at diagrammet tegnes på motsatt side enn først antatt.

## A.2.2 Skalering av resultatdiagrammer

ID	59
Klassifisering	Endring
Modifisert dato	04.02
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	04.02
Rapportert av	Torjus

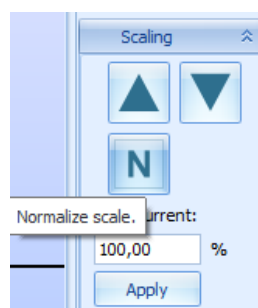
### Endringer

Skaleringen av resultatdiagrammer forblir uendret når det veksles mellom ulike resultatdiagrammer. Tidligere ble alle resultatdiagrammer normalisert til en standardskala ved veksling mellom visning av ulike diagrammer.

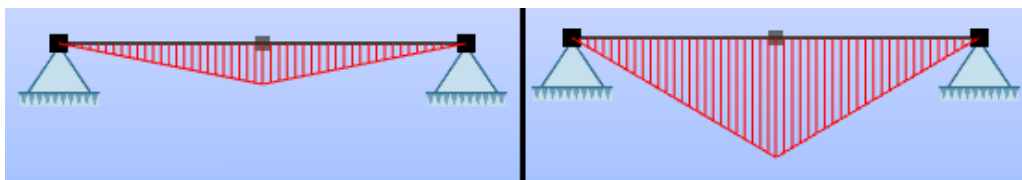
Skaleringen av diagrammene beholdes nå helt til brukeren benytter seg av skaleringsverktøyene i *Toolbox* eller lukker programmet.

Fjernet kallet til metoden `NormalizeResultScale()`.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmMainGui.ResultsPage



Figur A.41: Skaleringsverktøyene i *Toolbox*. *Normalize scale* tilbakestiller skalaen.



Figur A.42: Momentdiagram med normalisert skala t.v. og økt skalering t.h.

### A.2.3 Tegning av *joints* og *supports*

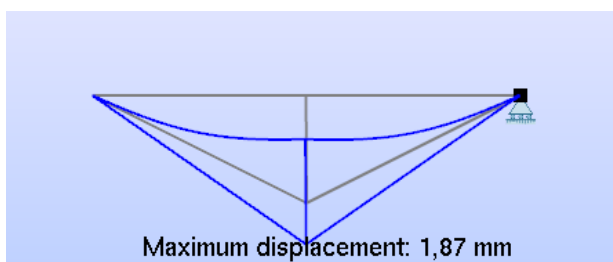
ID	22	Rapportert i versjon	3.0
Klassifisering	Feil	Rapportert dato	Høst 2013
Modifisert dato	31. januar	Rapportert av	Arne Aalberg
Modifisert dato	Espen	Endrede klasser	
<i>Namespace</i>		ComputationalModel.ModelTools.cs	
fap2D.Computations			

#### Feil

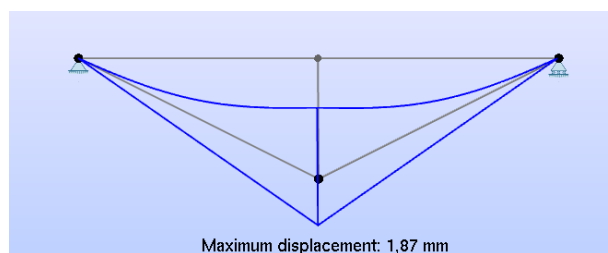
*Joints* og opplagere ble ikke alltid tegnet i resultatfanen. Typen opplager hadde ikke noe å si. Feilen gjorde det umulig å hente ut reaksjonskreftene.

**Endringer** Programmet tegner nå alle *joints* og opplagere. Metoden `ModelTools.GenerateMesh()` la kun til *joints* og medfølgende opplager dersom *jointet* tilhørte en *beam* eller *arch member*.

<i>Namespace</i>	Endrede klasser
fap2D.Computations	ComputationalModel.ModelTools.cs



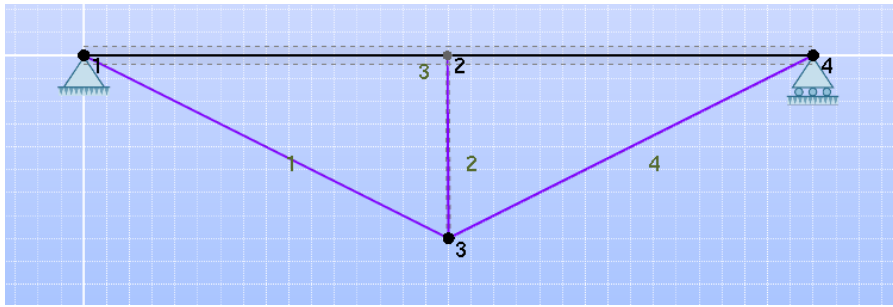
Figur A.43: Deformasjon av modell i figur A.45 før endring.



Figur A.44: Deformasjon av modell i figur A.45 etter endring.

#### Detaljer

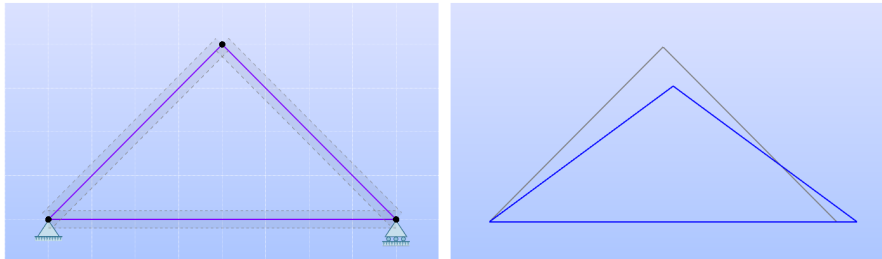
Studenter og tidligere brukere av **fap2D** har lagt merke til at *joints* og opplagere ikke alltid ble tegnet i resultatfanen. Figur A.43 viser at *joint* nummer 1, 2 og 3, og opplageret ved *joint* 1 har forsvunnet. Brukere påstod at typen opplager ikke hadde noe å si, i.e. både glidelager, fritt opplager og fast innspenning kunne forsvinne.



Figur A.45: Eksempel på konstruksjonsmodell med *bar* og *beam members*.

I den reproduerte konstruksjonsmodellen oppstod ikke alltid feilen, men i filer som ble tilsendt fra brukere oppstod den alltid.

Nummereringen av *memberne* virket å ha betydning. *Joints* som var tilknyttet en *bar/cable/truss member* ikke ble inkludert i beregningsmodellen dersom den aktuelle *memberen* hadde lavere ID enn andre *members* tilknyttet samme *joint*. Som vist i figur A.46 vises ingen *joints* når det kun er *bar members*.



Figur A.46: Enkelt fagverk der *joints* og opplagere forsvinner

Årsaken til feilen var ganske enkelt at *meshing*-metoden `GenerateMesh()` i `Computational-Model.ModelTools.cs` ikke la *joints* til beregningsmodellen dersom *jointet* ble opprettet av en *bar member*. Løsningen var å inkludere linjen:

```
_joints.Add(j);
```

i løkken som legger til *bar/cable/strut members*.

Det ble nødvendig å refaktorisere koden for å unngå å gjenta slike feil. Linjene som hentet ut *joints* fra konstruksjonsmodellen og la dem til i beregningsmodellen ble lagt i en ny metode `GetHelpNodeFromJoint()`.

#### A.2.4 Ulike farger for *bar*, *cable* og *strut members*

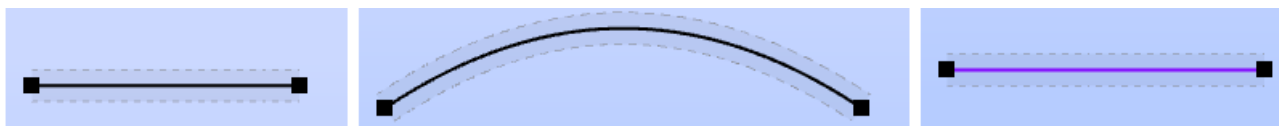
ID	40	Rapportert i versjon	3.0
Klassifisering	Nytt	Rapportert dato	November 2013
Modifisert dato	11. februar	Rapportert av	Torjus
Modifisert dato	Torjus		
<i>Namespace</i>	Endrede klasser		
fap2D.StructuralData	SubMember		
fap2d.Utilities	GlobalSettings		
	GlobalSettings.StructuralModel		

## Endringer

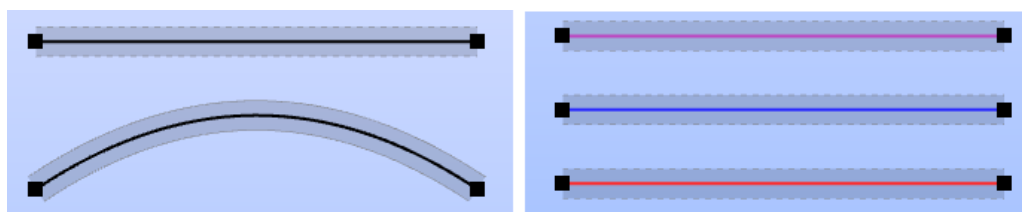
For å gjøre det enklere å skille de ulike typene *members* har *bar*, *strut* og *cable members* fått nye farger. Fargene gjenspeiler egenskapene til de ulike typene:

- Bars: Tar både strekk og trykk (lilla), som før.
- Struts: Tar kun trykk (blå).
- Cables: Tar kun strekk (rød).

Fargene er kun gjeldende i modellvinduet, for i resultatvinduet blir fargene på *members* alltid overskrevet av fargene som signaliserer strekk/trykk/forskyvning, også når det ikke er krefter i dem.



Figur A.47: Før endring: *Beam*, *arch beam* og *bar/strut/cable*.



Figur A.48: Etter endring: *Beam*, *arch beam*, *bar*, *strut* og *cable*.

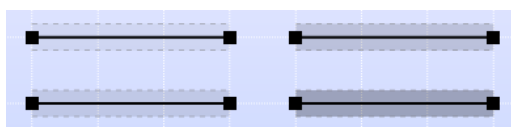
## A.2.5 Mer markant *member outline*

ID	62	Rapportert i versjon	3.0
Klassifisering	Modifikasjon	Rapportert dato	Vår 2013
Modifisert dato	26. februar	Rapportert av	Kolbein
Modifisert dato	Torjus		
<i>Namespace</i>		Endrede klasser	
fap2D.StructuralData		SubMember	
fap2D.Utilities		GlobalSettings	
		GlobalSettings.StructuralModel	

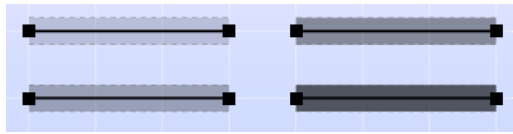
## Endringer

*Member outline* er gitt en mindre transparent farge enn før, slik at den blir mer tydelig: `CrossSectionColor = new Color4f(0.0f, 0.0f, 0.0f, 0.15f);`. Tidligere var transparenten satt til `0.05f`, altså 95 %.

Overlappende *members* markeres dermed også tydeligere enn før.



Figur A.49: Før endring: *Member outline* for hhv. 0, 1, 2 og 5 overlappende beams.



Figur A.50: Etter endring: Member outline for hhv. 0, 1, 2 og 5 overlappende beams.

### A.2.6 Fjernet *None/Ghost member*

ID	46	Rapportert i versjon	3.0
Klassifisering	Modifikasjon	Rapportert dato	Vår 2013
Modifisert dato	26. februar	Rapportert av	Kolbein
Modifisert dato	Torjus		
<i>Namespace</i>		Endrede klasser	
fap2D.StructuralData		Member	
fap2D.Utilities		GlobalSettings	
fap2D.Gui		GlobalSettings.StructuralModel	
		FrmMemberPropertiesDialog.Designer	

#### Endringer

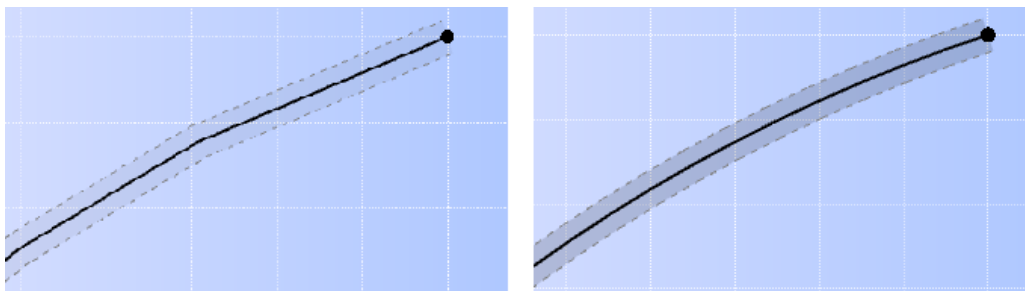
Har fjernet *None/Ghost member* fra programmet. Tidligere var denne brukt som en hjelpelinje, blant annet for å lage nye *joints* langs en *arch member*. *None member* har ikke lenger noen funksjon, fordi den nevnte problemstillingen behandles på en mer elegant og effektiv måte nå.

### A.2.7 Findelt tegning av *arch members*

ID	121	Rapportert i versjon	3.0
Klassifisering	Modifikasjon	Rapportert dato	Vår 2013
Modifisert dato	5. mai	Rapportert av	Espen, Kolbein
Modifisert dato	Espen		
<i>Namespace</i>		Endrede klasser	
fap2D.StructuralData		Member.cs	

#### Endringer

Endret antall tegnepunkter for *arch members* fra 30 til 90.



Figur A.51: t.v. grov inndeling av tegnepunkter for *arch members* - t.h. fin inndeling av tegnepunkter

## Detaljer

Beregningen av tegnepunkter til *OpenGL*-metodene var ikke tilstrekkelig nøyaktige nok til å tegne opp en jevn *arch member*.

Antall tegnepunkter per *arch member* ble endret fra 30 til 90. Bakgrunnen for valget av tegnepunkter er at det er en forbedring som visuelt virker tilstrekkelig i forhold til det gamle antallet. I tillegg tegnes *arch memberen* med linjestykker som tilsvarer omtrent 2° om gangen, altså nesten ett linjestykke per grad.

Det bør også nevnes at punktene ble beregnet langs korden, antakelig ved en feil, noe som gjorde at linjestykkene som lå på endene av *arch memberen* ble lengre enn de andre.

## A.2.8 Komplett *grid*

### Feil

De ytterste linjene i rutenettet (*grid*) blir ikke tegnet. Brukeren merker ikke det når *grid* er stort, men når det er lite merkes det.

### Endring

Rutenettet tegnes nå som et fullstendig nett.

### Argumentasjon

Endringen var enkel å utføre, men ikke særlig viktig.

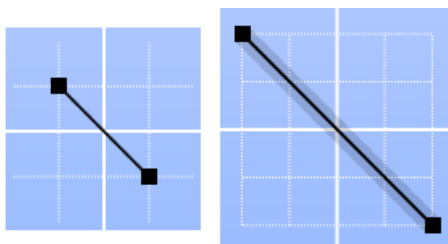
### Metadata

ID:	95
Klassifisering:	Feil, Endring
Sist modifisert:	12. mai
Modifisert av:	Torjus
Rapportert i versjon:	3.0
Rapportert dato:	Vår 2014
Rapportert av:	Torjus

### Endrede klasser

fap2D.Gui	FrmSpringDialog
fap2D.Utilities	Utilities

### Figurer



Figur A.52: **3.0** (t.v): Rutenettet mangler de ytterste linjene. **3.0** (t.h): Rutenettet tegnes komplett.

## A.2.9 Bruk av færre linjestykker ved tegning av sirkler

### Endring

Redusert antall linjestykker som brukes for å tegne opp sirkler, fra 100 til 70. Sirkler tegnes nemlig opp vha rette linjestykker, så det trengs en viss mengde linjer (men altså ikke for mange).

### Argumentasjon

70 linjer er mer enn nok for å tegne en glatt sirkel, selv for store sirkler. Siden tegnemethoder kjøres kontinuerlig i programmet sparer man inn litt ressursbruk, selv om endringen kan regnes som ubetydelig i nyere PC-er.

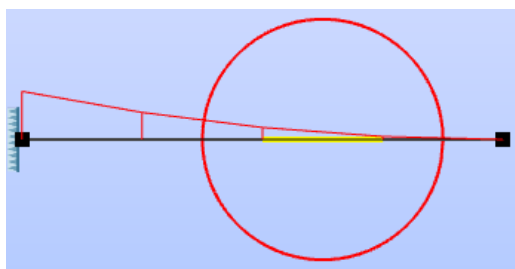
### Metadata

ID:	103
Klassifisering:	Endring
Sist modifisert:	14. mai
Modifisert av:	Torjus
Rapportert i versjon:	3.0
Rapportert dato:	Vår 2013
Rapportert av:	-

### Endrede klasser

fap2D.Gui	UC_CouplingSpring
	UC_JointProperties
	UC_Release
fap2D.StructuralData	Release
fap2D.Computations	Element

### Figur



Figur A.53: 3.1: Tegning av sirkel vha. 70 linjestykker.

## A.3 Markering av *structural items*

### A.3.1 Feilaktig *snap* ved museklikk

ID	49	Rapportert i versjon	3.0.0 - Beta
Klassifisering	Feil	Rapportert dato	21.02
Modifisert dato	09.05	Rapportert av	Petter H. Holmstrøm
Modifisert dato	Torjus		

### Feil

*Joints* som ikke er plassert i hjørnepunkter i rutenettet (*grid*) vil flyttes til nærmeste hjørnepunkt dersom man markerer *joint*-et eller tilknyttede *members* når *Snap to grid* er aktivert. Dette betyr at man er nødt til å deaktivere *Snap to grid* hver gang man skal markere et slikt *joint/member*.

Grunnen til feilen er at all bruk av musen i **fap2D** styres av et sett med *Mouse events*. Disse *events* vil alltid tolke et museklikk som et potensielt flytt av musen fordi selv de minste musebevegelser

registreres mellom man trykker ned og slipper opp en musetast. Dermed kjøres flytte-metodene i programmet også for museklikk. Når *Snap to grid* er aktivert får vi derfor en uønsket reaksjon.

### Endringer - Versjon 3.3.0

For å unngå at *joints* og *members* feilaktig flyttes når intensjonen er å bare markere dem er det lagt til en prosedyre som hindrer flytting.

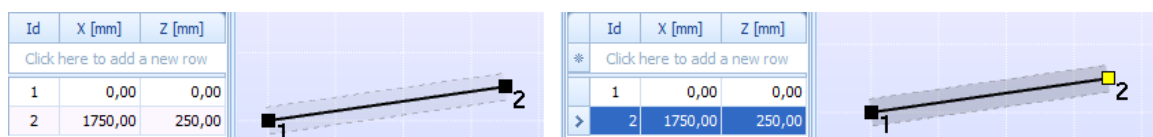
Alle flytt av *joints/members* som er kortere enn 10 % av minste *x-* eller *z-* *grid spacing* vil oppfattes som markeringer selv om *Snap to grid* er aktivert. Markerte *joints/members* vil ikke bli flyttet.

For å rette feilen måtte metoden `MoveStructuralItems(...)` endres. I tillegg måtte lengden på flyttene som gjøres registreres i *joint data*-tabellen (`FrmMainGui.LeftPanel`) og i `FrmModel`.

Namespace	Endrede klasser
fap2D.Gui	FrmMainGui FrmMainGui.LeftPanel FrmModel



Figur A.54: Før endring. Ved markering av *joint 2* blir det flyttet til et hjørne i *grid*.



Figur A.55: Etter endring. Ved markering av *joint 2* blir posisjonen uforandret.

### Detaljer - Versjon 3.3.0

Feilen er som flere andre feil med på å gi brukeren en følelse av å ikke være i kontrollsetet. Den kan endre geometrien i en modell uten at bruker oppdager det (dersom *joints* er lagt nært opptil hjørnepunktene, men ikke helt opptil). Derfor ble feilen høyt prioritert.

### Endringer - Versjon 3.5.0

Kriteriet som brukes for å skille mellom flytt og museklikk er endret i forhold til i forrige versjon. I stedet for å bruke et kriterium som avhenger av hvor langt et flytt er relatert til *grid spacing*, brukes nå et kriterium basert på *screen coordinates*:

Dersom et *structural item* flyttes mindre enn 28 piksler langs skjermen vil det ikke bli flyttet når musepekeren slippes.



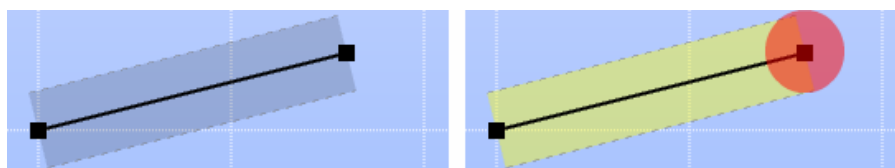
### Detaljer - Versjon 3.5.0

Det nye kriteriet, 28 piksler, er knyttet til en innstilling ved navn `MinMouseMoveDist` i koden.

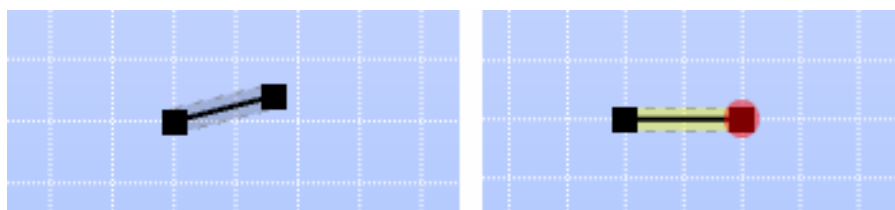
Grunnen til endring av kriterium er at *screen coordinates* er statiske. Derfor vil et kriterium basert på et slikt koordinatsystem passe godt, uavhengig av zoom-nivå og *grid spacing*.

Et kriterium basert på *grid spacing* vil variere ut i fra *grid spacing* og zoom-nivå. Når det zoomes langt ut vil derfor ikke et slikt kriterium fungere, fordi avstanden mellom linjene i rutenettet på skjermen blir liten. Dermed kan et museklikk i verste fall bli tolket som et flytt.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmModel



Figur A.56: Dersom man zoomer **langt nok inn** vil et museklikk **ikke** føre til uønsket *snap* (når *world coord.*-kriterium brukes). Den røde sirkelen illustrerer flytt-kriteriet. Sirkelen er overdrevet for å fremheve poenget.



Figur A.57: Dersom man zoomer **langt ut** vil et museklikk føre til uønsket *snap* (når *world coord.*-kriterium brukes). Den røde sirkelen illustrerer flytt-kriteriet, som nå er farlig lite.

### A.3.2 Markering av *members* med eksentrisitet

ID	98	Rapportert i versjon	3.3.0 - DE
Klassifisering	Feil	Rapportert dato	13.05
Modifisert dato	21.05	Rapportert av	Espen
Modifisert dato	Espen		

#### Feil

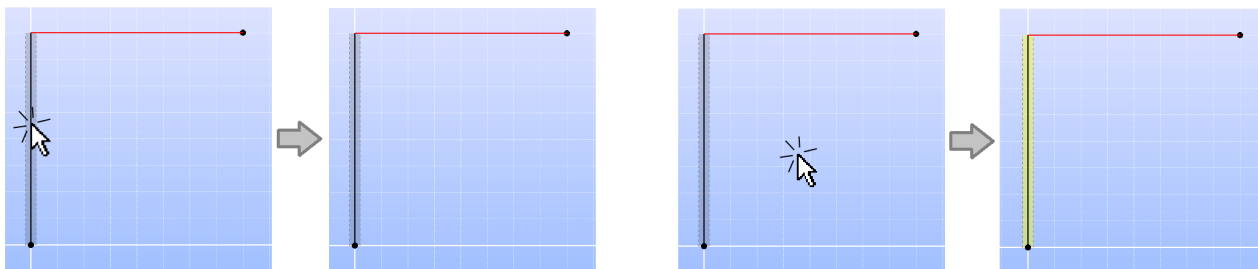
Dersom man hadde en *member* med eksentrisiteter ble *memberen* tegnet korrekt med endene justert for eksentrisiteter, mens området man måtte trykke på for å velge den var basert på posisjonen før eksentrisiteter ble lagt på.

Feilen var at metoden som søkte etter *member* på den aktuelle pekerposisjonen antok at posisjonen til enden av *memberen* var den samme som posisjonen til *jointet*. Dette er ikke tilfellet dersom *memberen* har eksentrisiteter.

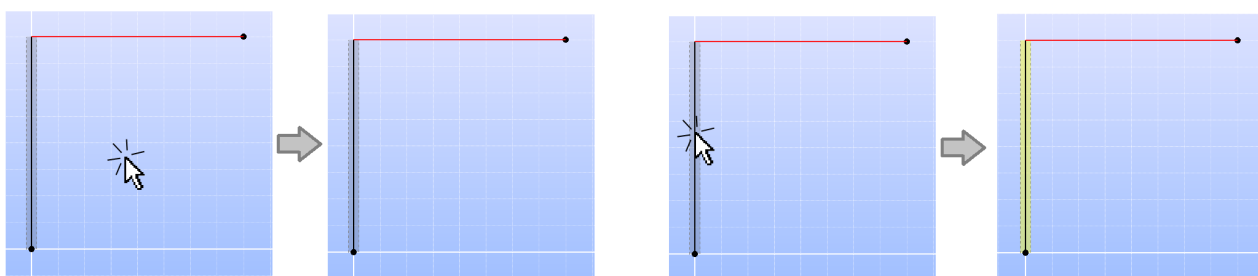
## Endringer

Rettet feilen ved å endre hvilke punkter `ContainsCoordinates(...)` i `SubMember.cs` bruker til å beregne markering til `PointEnd1` og `PointEnd2` som tar hensyn til eksentrisiteter.

<i>Namespace</i>	Endrede klasser
fap2D.StructuralData	SubMember.cs



Figur A.58: Før - ved markering tror pekeren at *memberen* er mellom *jointene*.



Figur A.59: Etter - korrekt markering av *memberen*.

### A.3.3 Visning og markering av forskyvde elementer

ID	56	Rapportert i versjon	3.0
Klassifisering	Nytt	Rapportert dato	Vår 2014
Modifisert dato	15. mai	Rapportert av	Petter H. Holmstrøm
Modifisert dato	Torjus, Espen		
<i>Namespace</i>		Endrede klasser	
fap2D.Gui		FrmModel	
fap2D.Computations		FrmModel.OpenGL	
		Element	

### A.3.4 Markering av *submembers*

ID	75	Rapportert i versjon	3.0
Klassifisering	Modifikasjon	Rapportert dato	Høst 2013
Modifisert dato	15. mai	Rapportert av	Kolbein
Modifisert dato	Torjus, Espen		
<i>Namespace</i>		Endrede klasser	
fap2D.StructuralData		SubMember	

#### Feil

Det har vært vanskelig å markere (*sub*)*members* i modellvinduet fordi metoden som brukes for å vurdere om et museklikk “treffer” en *submember* har krevd at man trykker veldig nær senterlinjen i *submember*-en.

#### Endring

Derfor er metoden endret slik at dersom man trykker innenfor *member outline*, altså innenfor tverrsnittshøyden, så vil *submember*-en markeres som valgt. Endringen gjelder så langt kun for rette *submembers*. Metoden `ContainsCoordinates(...)` benytter nå lokal *member*-definisjon som kriterium for markering (globalt system før)

For *arch members* er det lagt til en ny innstilling som kan brukes til å gjøre markeringsområdet rundt *arches* større (avstanden fra senterlinjen i piksler).

### A.3.5 Markering av elementer

ID	78	Rapportert i versjon	3.0
Klassifisering	Feil, Modifikasjon	Rapportert dato	Høst 2013
Modifisert dato	15. mai	Rapportert av	Espen, Torjus, ...
Modifisert dato	Torjus		
<i>Namespace</i>		Endrede klasser	
fap2D.Computations		Element	

### A.3.6 Markeringspreferanse

ID	118	Rapportert i versjon	3.0
Klassifisering	Modifikasjon	Rapportert dato	Høst 2013
Modifisert dato	7. mai	Rapportert av	Kolbein
Modifisert dato	Torjus		
<i>Namespace</i>		Endrede klasser	
fap2D.Gui		FrmModel	

#### Endring

Markeringspreferansene i modellvinduet i **fap2D** er endret.

### Markeringspreferanse i versjon 3.0:

- *Modelling display*
  1. *joints (og internal joints)*
  2. ***members (og submembers)***
  3. *point loads*
  4. *line loads*
- *Results display*:
  1. *Joints (og internal joints)*
  2. Elementer

### Markeringspreferanse i versjon 3.1:

- *Modelling display*
  1. *joints (og internal joints)*
  2. *point loads*
  3. *line loads*
  4. ***members (og submembers)***
- *Results display*:
  1. *Joints (og internal joints)*
  2. Elementer

### Argumentasjon

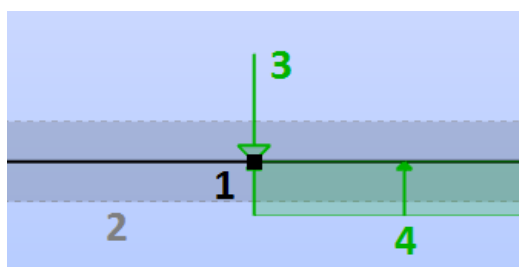
Grunnen til endringen er at det nå er mulig å markere (*sub*)*members* mye enklere enn før fordi de kan markeres innenfor *member outline*. Se vedlegg A.3.4 for detaljer. Dette gjør samtidig at det blir vanskeligere å markere *point loads* og *line loads* fordi de vil overlape mer med (*sub*)*members*. Siden det alltid er mulig å markere (*sub*)*members* på den siden det ikke vises symbol for *line load* (se eksempel i figur A.61) anses løsningen fortsatt som passende for alle *items*. De minste *items* har høyest preferanse, og de største lavest preferanse, slik at der mulig å filtrere ut alle *items*.

### Detaljer

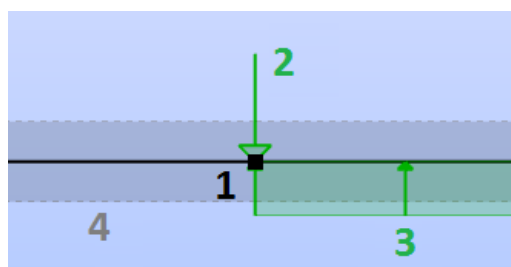
Logikken som brukes i kildekoden for å registrere markeringer baserer seg på vektorregning, og kan forklares slik:

1. Brukeren trykker med musepekeren et sted inne i modellvinduet.
2. Programmet registrerer hvor i panelet brukeren har trykket, angitt i *screen coordinates*.
3. Det kjøres en prosedyre for hvert objekt på skjermen som avdekker om trykket som ble gjort skal tolkes som en markering av objektet.
4. Dersom flere objekter blir tolket som markerte vil ett av dem velges, avhengig av hvilken preferanse de har. *Joints* har høyest preferanse både i versjon 3.0 og 3.1.
5. Prosedyren har nå valgt det markerte objektet og vil utføre videre handlinger basert på det, f.eks. åpne en dialogboks.

### Figurer



Figur A.60: 3.0: Den gamle markeringspreferansen.



Figur A.61: 3.1: Den nye markeringspreferansen. *Member*-en til høyre for *joint*-et kan markeres ved å trykke på oversiden av senterlinjen.

## A.4 Konstruksjonsmodell

### A.4.1 Validering av *arch member*-geometri

ID	2
Klassifisering	Feil
Modifisert dato	27.05
Modifisert av	Torjus
Rapportert i versjon	3.0.0 Beta
Rapportert dato	August 2013
Rapportert av	Torjus, Kolbein Bell

#### Feil

Det er mulig å lage *parabolic arch members* som ikke er realistiske å bruke i analysene i fap2D fordi forholdet mellom *height* og *chordlength* kan settes tilnærmet uendelig høyt. Dette vil føre til at toppen av parabelen blir veldig krapp/spiss og man vil i realiteten få spenningskonsentrasjoner i toppen i hht. bruddmekanisk teori. **fap2D** er ikke laget for å analysere slike konstruksjoner.

Det er mulig å flytte *joints* som er tilknyttet en *circular arch member* slik at *member*-en ikke lenger blir gyldig definert. Dette skjer når man prøver å sette avstanden (*chordlength*) mellom de to *joints* i *member*-en til større enn:  $2 \cdot \text{radius}$  (*height*).

Det er mulig å endre radiusen (*height*) i en *circular arch member* slik at *member*-en ikke lenger blir gyldig definert. Dette skjer når man prøver å sette radius (*height*) til mindre enn  $\text{chordlength}/2$ , via *member properties*-dialogen.

#### Endringer - Versjon 3.3.0 Developer

For å hindre at *parabolic arch members* blir urealistiske vises det en advarsel dersom brukeren oppretter en *parabolic arch member* eller flytter *joints* i en *parabolic arch member* slik at forholdet mellom *height* og *chordlength* blir større enn 3.

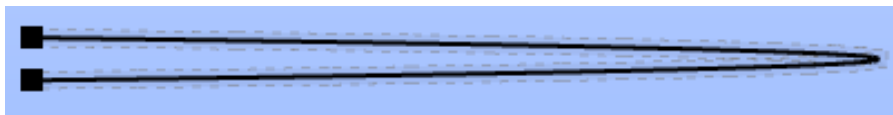
Dersom brukeren prøver å lage en ny *parabolic arch member*, vha. musepekeren, der det nevnte forholdet blir større enn 10 så blir *member*-en slettet og brukeren varslet.

Dersom brukeren prøver å flytte *joints* som er tilknyttet en *parabolic arch member* slik at det nevnte forholdet blir større enn 10 så vil *member*-en bli hindret fra å flyttes.

Brukeren varsles enten via en advarsel i *error list*, eller via en dialogboks dersom endringen gjøres fra *joint properties*- eller *member properties*-dialogene.

<i>Namespace</i>	Endrede klasser
fap2D.StructuralData	Joint Member
fap2D.Gui	FrmModel FrmJointPropertiesDialog FrmMemberPropertiesDialog FrmMainGui FrmMainGui.LeftPanel

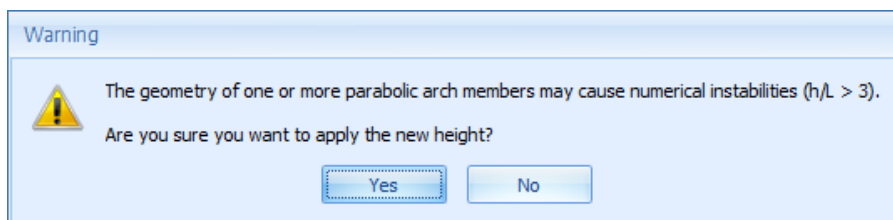
### Figurer - *Parabolic arch members*



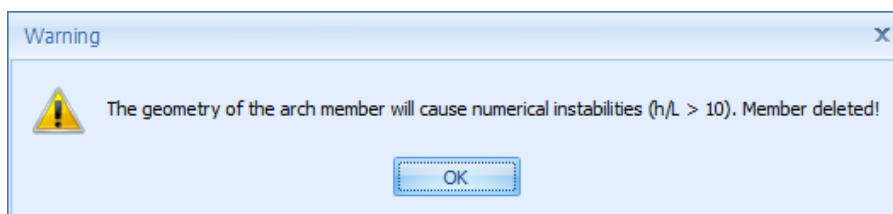
Figur A.62: *Parabolic arch member* som er veldig krapp/spiss i toppen.

Error List		
Error...	Description	Time stamp
	The geometry of one or more parabolic arch members will cause numerical instabilities ( $h/L > 10$ ). The joint will not be moved.	14:33:02
	The geometry of the arch member may cause numerical instabilities! ( $h/L > 3$ )	14:31:39

Figur A.63: Etter endring: Advarsler i *error list* pga. krappe/spisse *parabolic arch members*.



Figur A.64: Etter endring: Dialog med advarsel om at *height/chordlength* > 3. Oppstår når man oppretter en krapp/spiss *parabolic arch member* vha. musepekeren.

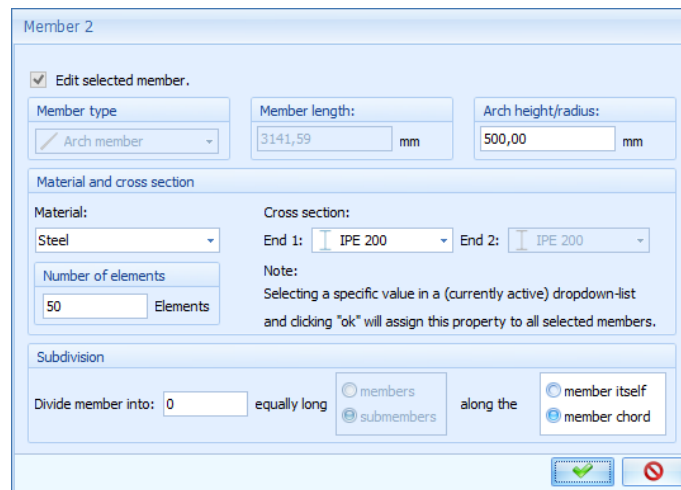


Figur A.65: Etter endring: Dialog med advarsel om at *height/chordlength* > 10. Oppstår når man oppretter **for** krapp/spiss *parabolic arch member* vha. musepekeren.

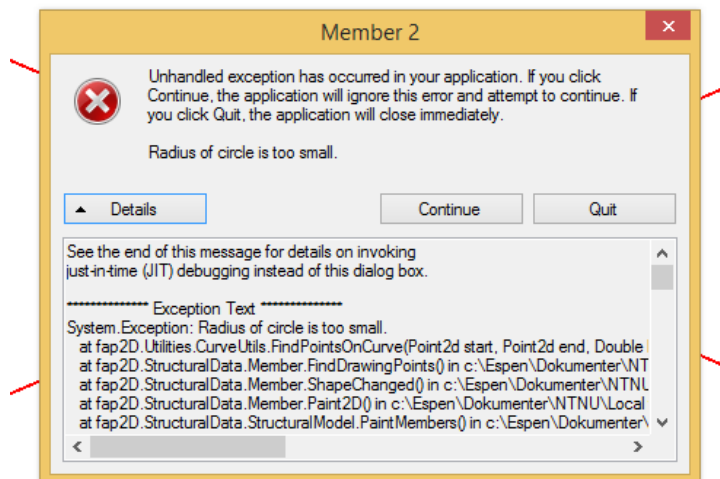
### Figurer - *Circular arch members*

Id	X [mm]	Z [mm]
Click here to add a new row		
1	0,00	0,00
2	2000,00	0,00

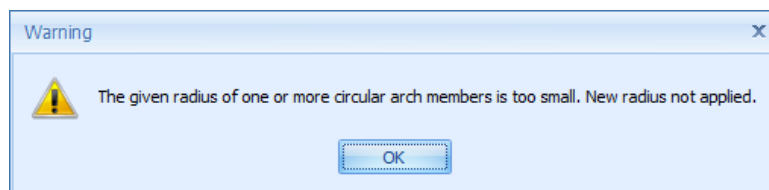
Figur A.66: Før endring: Veldefinert *circular arch member* med radius 1000 mm.



Figur A.67: Før endring: Radius for *circular arch member* endres til 500 mm.



Figur A.68: Før endring: Resultat av ny radius for *circular arch member* er et ugyldig forhold mellom *height* og *chordlength* som igjen fører til en *unhandled exception*.



Figur A.69: Etter endring: Dialog med varsel om at radius i *circular arch member* blir for liten.

### Endringer - Versjon 3.5.1 Developer

I forrige versjon ble det glemt å ta hensyn til at *arch members* kan flyttes vha. eksentrisiteter i endene. I **3.5.1 Developer** er det nå tatt hensyn til dette i *eccentricity*-dialogen, på samme måte som for *joint properties*-dialogen.

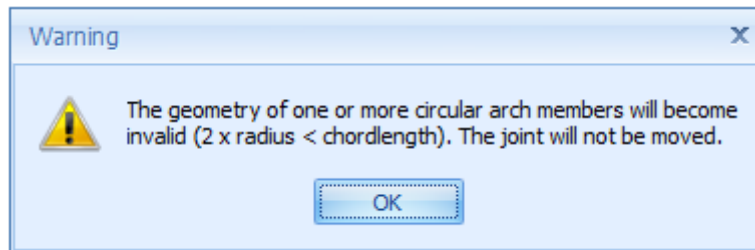
Denne rettingen henger sammen med rettingene som er gjort i punkt 128 XXX ref her.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmEccentricityDialog

### Endringer - Versjon 3.1

En liten feil i *member properties*-dialogen i versjon **3.3.0 Developer** gjorde at valideringen av *arch members* ikke fungerte. Den er nå fikset.

Advarselen i dialogboksen i figur A.69 ble endret fordi den var upresis:



Figur A.70: Mer presis advarsel enn i figur A.69.

I forbindelse med at det i versjon **3.3.0 Developer** ble innført begrensninger og advarsler for *arch members* er det lagt til en innstilling som gjør at brukeren kan overse advarslene. Se punkt 109 XXX ref her for detaljer.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmEccentricityDialog FrmMemberPropertiesDialog FrmJointPropertiesDialog FrmModel FrmainGui.LeftPanel

#### A.4.2 Deling av *arch members*

ID	53
Klassifisering	Feil
Modifisert dato	26.02
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	September 2013
Rapportert av	Kolbein Bell

### Feil

Ved inndeling av *arch members* i *submembers* blir ikke *submembers* alltid like lange. Dette gjelder ved deling langs *member itself*. Ulike kombinasjoner av odde/like antall *submembers* og *circular/parabolic arch* gir tilsynelatende vilkårlige lengder.

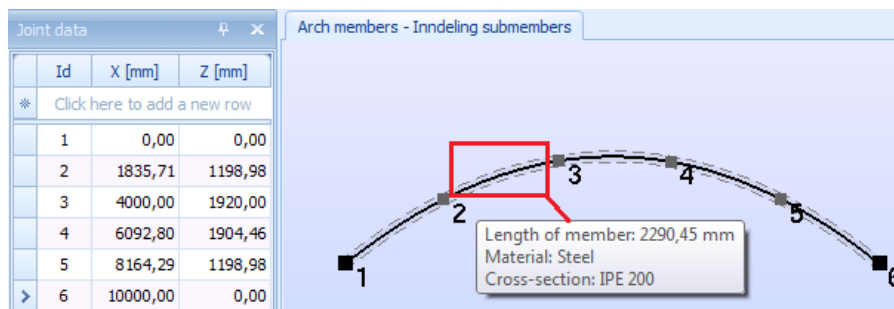


## Endringer

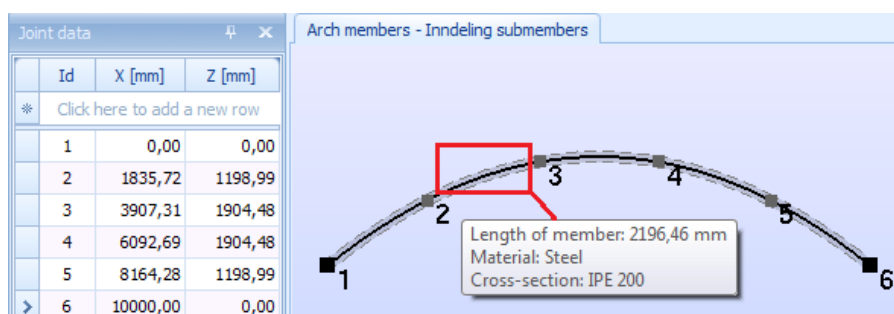
Feilen er håndtert ved å legge til et ekstra argument, `bool explicitDivison`, i metoden `Divide(double x, double z)` i klassen `SubMember`. Argumentet sørger for at kallet til `GetClosestPoint(double x, double z)` ikke gjennomføres ved eksplisitt inndeling i nye *sub-members* (deling vha. *member properties*-dialogen). I tillegg er kallet til `GetClosestPoint(double x, double z)` fjernet fra en konstruktør i klassen `Joint`, fordi det var overflødig.

Inndelingen gir nå like lange elementer uansett hvilken konfigurasjon av *arch members* som brukes. Ved implisitt deling (et *joint* plasseres på eksisterende *member* og forårsaker deling) blir `GetClosestPoint(double x, double z)` kalt som før.

<i>Namespace</i>	Endrede klasser
fap2D.StructuralData	Member SubMember Joint
fap2D.Gui	FrmModel FrmMemberPropertiesDialog
fap2D.Computations	ComputationalModel.LoadTrain



Figur A.71: Før endring.



Figur A.72: Etter endring.

### A.4.3 Snap av circular arch members

ID	33
Klassifisering	Feil
Modifisert dato	28.01
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	September 2013
Rapportert av	Espen, Torjus

#### Feil

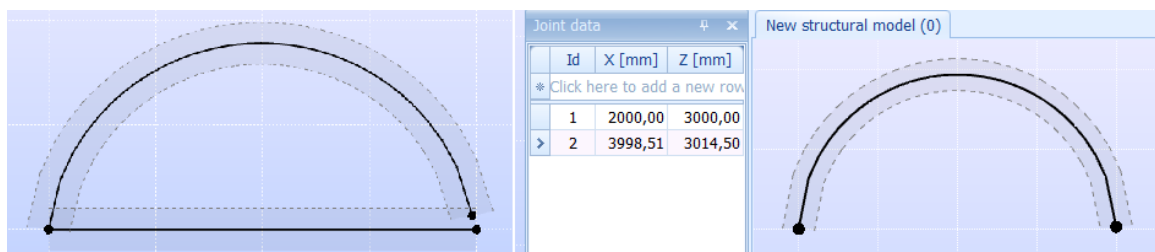
*Circular arch members* kunne ikke *snappe* til grid slik at den blir en full halvsirkel, dvs. kordelengden<sup>2</sup> er lik  $2 \cdot radius$ . Dersom *arch memberen* klarer å *snappe* til eksisterende *joint* får man feilmelding om at radius er for liten.

#### Endringer

Metoden `FindShortestDistanceToCurve(...)` i `CurveUtils.cs` kalles nå med en dekkende `try/catch`-blokk, og parameterne som sendes inn er verdier som tilsvarer modellen, ikke skjermkoordinater. Metoden `JointChanged(...)` i `Joint.cs` tillater nå *circular arch member* med full utstrekning, dvs.

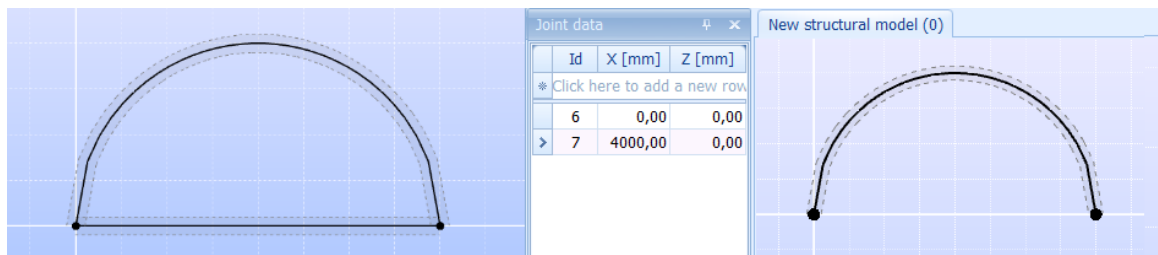
$$L_{korde} \leq 2 \cdot radius$$

Namespace	Endrede klasser
fap2D.StructuralData	Member SubMember
fap2D.Utilities	CurveUtils
fap2D.Computations	ComputationalModel.ModelTools Element ComputationalModel.Drawing ComputationalModel



Figur A.73: Før endring.

<sup>2</sup>avstanden mellom endejointene på en arch member



Figur A.74: Etter endring.

## Detaljer

*Arch members*, definert av radius, med fullt utstrekning klarer ikke å benytte *snap* funksjonen.

### Del 1:

Når man trekker en *circular arch member* stanser *memberen* å strekke seg ut like før den når full utstrekning. Prøver man å flytte et av *jointene* ved å endre koordinatene flytter *jointet* seg tilbake.

Metoden `JointChanged(...)` kalles hver gang at *joint* har beveget seg, dvs. flere ganger mens en *member* settes ut. Når man lager eller flytter det andre *jointet* av en *circular arch member* skjer det en beregning som sjekker at *jointene* som definerer *arch memberen* ikke er for langt fra hverandre. Denne begrensningen var satt til å være for streng. Dersom kordelengden var større eller lik  $2 \cdot radius$ , ville *jointet* som beveget seg flyttes tilbake til en tidligere posisjon. Dette er nå endret til å skje kun dersom kordelengden er større enn  $2 \cdot radius$ :

```
private void JointChanged(...)
...
if (Geometry == DrawableItemGeometry.Circular
    && (PointEnd1-PointEnd2).Length > Height * 2)
...

```

### Del 2:

En feilmelding dukker opp dersom man snapper *ende-jointene* til eksisterende *joints* slik at *arch memberen* får full utstrekning. Feilmelding man får er "*illegal radius for circular curve(too small)*" og kastes av metoden `FindShortestDistanceToCurve(...)`.

Metoden `FindShortestDistanceToCurve(...)` er sensitiv forholdet mellom radius og kordelengde. Metoden kaster en *exception* dersom radiusen blir for liten, eller kordelengden blir for stor. Før kallet til metoden ble posisjonene til hvert *ende-joint* omregnet fra modellkoordinater til skjermkoordinater, og deretter fra skjermkoordinater tilbake til modellkoordinater:

```
public bool ContainsCoordinates(...)
...
leftPoint =
    OpenGLUtils.ConvertWorldToScreen(this.HostMember.Joint1.Position);
...
CurveUtils.FindShortestDistanceToCurve(
    OpenGLUtils.ConvertScreenToWorld(leftPoint),...)
...

```

Dette ble trolig gjort fordi pekerkoordinatene av og til ble brukt av metoden og måtte også omregnes. Metodene `OpenGLUtils.ConvertWorldToScreen(...)` og `OpenGLUtils.ConvertScreenToWorld(...)` gir ikke eksakte verdier fordi de beregner mellom piksler og teoretisk rom. Omregningen som skjer

vil bli tilnærmet og kan gi en feil som vanligvis rettes opp av av *snap* eller eksakt inntasting fra bruker.

Metodekallet ble endret til å fjerne stegene med omregning:

```
public bool ContainsCoordinates(...)
...
CurveUtils.FindShortestDistanceToCurve(this.HostMember.Joint1.Position,...)
...
```

#### A.4.4 Arch member med negativ radius

ID	114
Klassifisering	Endring
Modifisert dato	
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	September 2013
Rapportert av	

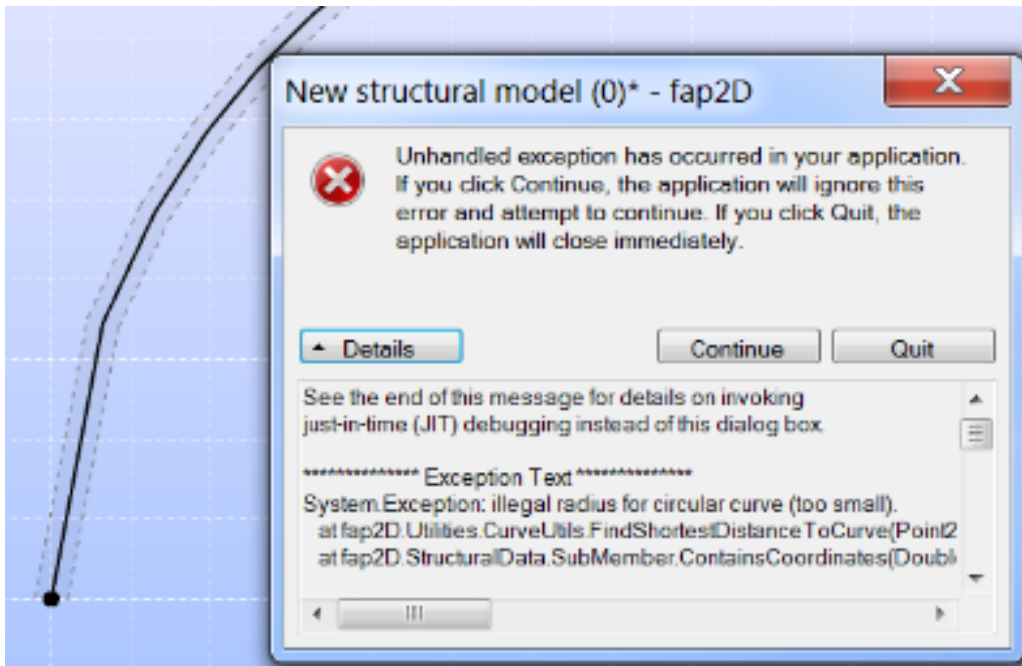
#### Endringer

Versjon 3.3.0 – DE: Fjernet mulighet for negativ høyde og radius for *arch members* fordi programmet ofte fikk feilmeldinger og krasjet da det ble brukt.

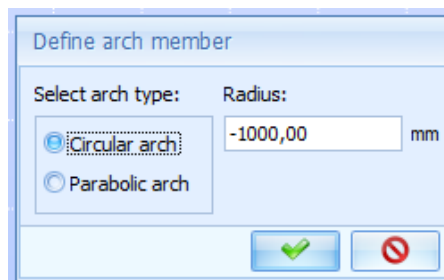
Versjon 3.5.0 – DE: Gjeninnførte negativ høyde og radius for *arch members*.

Alle tidligere oppdagede feil i sammenheng med *arch members* ble fikset slik at man ikke lenger opplever programkrasj.

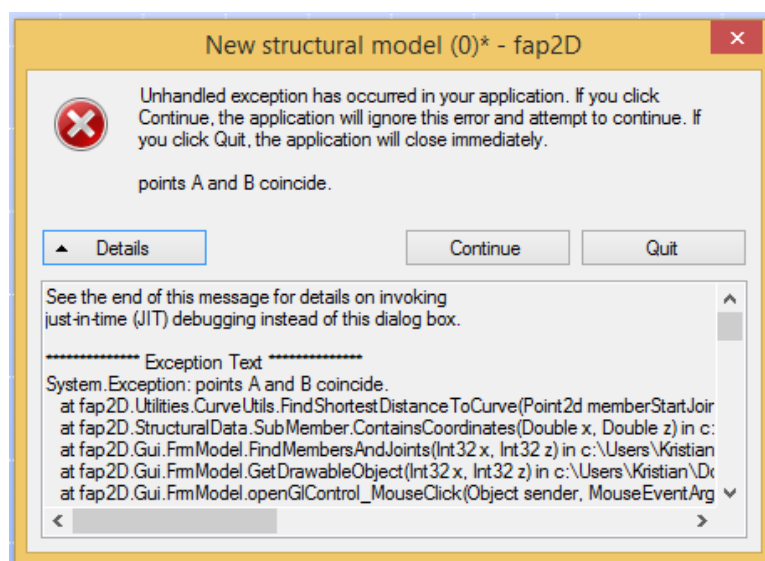
<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmArcMemberDialog.cs FrmMemberPropertiesDialog.cs
fap2D.Computations	ComputationalModel.LoadTrain.cs
fap2D.StructuralData	Joint.cs Member.cs SubMember.cs



Figur A.75: 3.0: Unhandled exception for circular arch member.



Figur A.76: 3.0: Angir negativ radius.



Figur A.77: 3.0: Unhandled exception: "points A and B coincide" ved bruk av negativ radius.

## Detaljer

### Versjon 3.3.0 - DE:

Feilen som førte til fjerning av negativ høyde og radius: Dersom man ga en *circular arch member* en negativ radius og prøvde å tegne den opp stoppet programmet med en *unhandled exception: "Points A and B coincide"*.

Programmet er begrenset slik at det ikke lenger er mulig å legge inn negativ høyde eller radius for *arch members*. Det innebærer at man må tegne *arch members* som krummer nedover ved å tegne *memberen* fra høyre mot venstre, med positiv høyde eller radius. Løsningen er ikke optimal, så det er mulig den bør forbedres.

### Versjon 3.5.0 - DE:

Opprinnelig var det mulig å ha *arch members* med negativ høyde/radius, men problemer og alvorlige feilmeldinger eller programkrasj gjorde at denne muligheten ble fjernet. Dette førte til at man ble nødt til å bruke alternative måter for å modellere hva man ønsket, og det ble etter hvert flere grunner til å gjeninnføre det.

Grunner for å ha mulighet for negativ radius på *arch members*:

- intuitivt – det er lett å forstå *arch members* med negativ høyde og radius
- ønske – det er flere ønsker om å ha muligheten istedenfor alternativer
- nødvendig dersom *members* skal være justert fra venstre mot høyre

For å legge opp til at *members* kun skal være definert fra venstre mot høyre, slik det er gjort i A.4.14 *Member*-retning, er man nødt til å ha muligheten for negativ radius. Den alternative måten å tegne en hengende *arch member* var å tegne *arch memberen* fra høyre mot venstre.

Temaet om tegning av *arch members* som krummer nedover ble tatt opp i spørreundersøkelsen som ble besvart av studenter som brukte **fap2D** i faget *Konstruksjonsmekanikk beregningsmetoder* våren 2014. Studentene ga inntrykk av at det mest intuitive er å benytte negativ radius til å tegne slike *arch members*. Se diskusjon i vedlegg B.

Årsaker til programkrasj tidligere var ofte forbundet med metoden `FindShortestDistanceToCurve(...)`. Mange av disse feilene var generelle og påvirket også rette *members*. Disse feilene ble rettet opp separat. Se også:

- A.4.11 *Members* med joints i samme punkt - *Arch members* med sammenfallende *joints* feilet i alle FORTRAN-rutinene for *arch members*.
- A.4.3 *Snap* av *circular arch members* - Koordinatene til endepunktene av *arch members* ble omregnet til å gjøre radius for liten i forhold til kordelengden

Andre FORTRAN-rutiner forbundet med *arch members* ble undersøkt. Kun `FindPointsOnCurve()` (*wrapper* metode for en tilsvarende FORTRAN-rutine) hadde eksplisitt dokumentert at den håndterte negativ radius, men ifølge K. Bell var alle rutinene implementert så generelle som mulig. Senere bruk av *arch members* med negativ radius tilsier at FORTRAN-rutinene takler det.

For å gjeninnføre negativ radius måtte alle metoder for validering av *arch members* endres til å validere absoluttverdien av høyde/radius.

I dialogboksene der det kan skrives inn verdi på høyde/radius måtte kontrollene endres til å godta negativ radius.

### A.4.5 Feilplassering av elementer med eksentrisiteter

ID	111
Klassifisering	Feil
Modifisert dato	01.04
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	21.03
Rapportert av	K. Bell

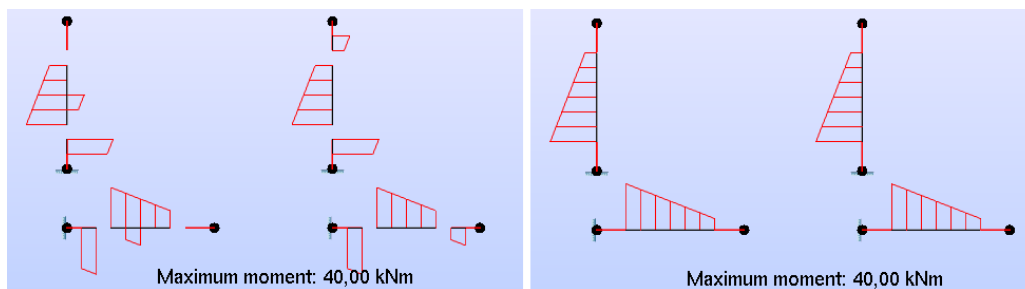
#### Feil

Feilen ble oppdaget ved at det ble brukt eksentrisitet i hver ende av en *member*. Diagrammer viste at elementer, spesielt på endene, ble tegnet med feil koordinater, og med feil vinkel.

#### Endringer

Feilen ble rettet opp ved å faktorisere beregningene av koordinatene for et element med eksentrisitet. Slik ble alt sentralisert og konsistent. Det ble da nødvendig å skille mellom to definisjoner av koordinater, én der eksentrisiteter var inkludert, og én uten.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmEccentricityDialog.cs
fap2D.Computations	Element.cs ComputationalModel.ModelTools.cs
fap2D.StructuralData	Eccentricity.cs



Figur A.78: Momentdiagram for modell før og etter – de venstre modellene i før og etter bruker eksentrisitetstypen *MoveMember*, de høyre modellene i før og etter bruker eksentrisitetstypen *MoveJoint*.

#### Detaljer

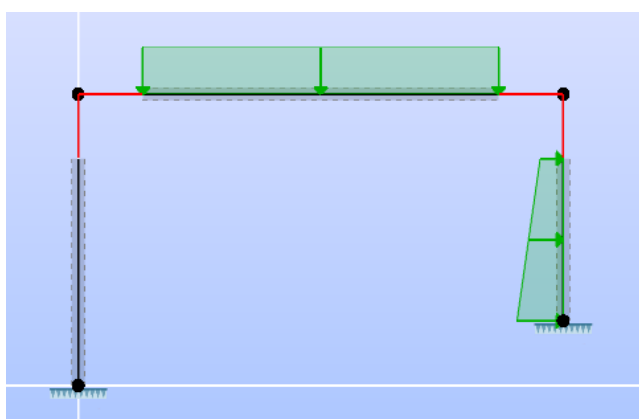
Elementene var mulig å markere, og hadde i mange tilfeller korrekte verdier, bare feil retning. Forskyvninger ble tegnet tilsynelatende korrekt. Dette var nok på grunn av at det ble tegnet med eksplisitte koordinater, og ikke et startpunkt, lengde og vinkel.

Eksentrisiteter ble satt på feil i tidligere implementasjon. Mange ulike kombinasjoner av orientering og typer gjorde at håndteringen av eksentrisitetene, som ble gjort flere ulike steder, ble implementert feil. Feilen gjaldt mest for elementer, men også for *submembers*.

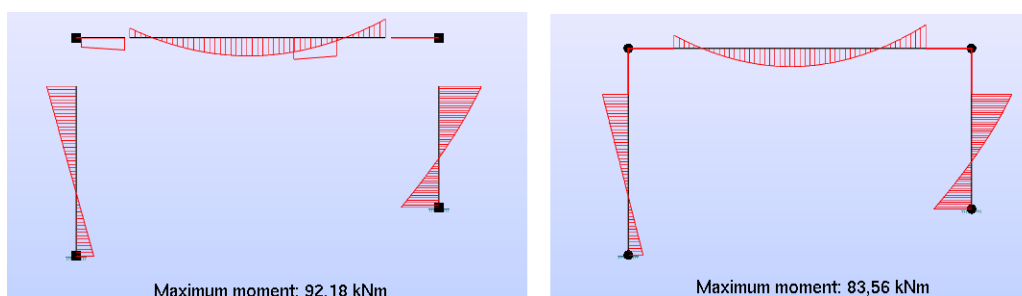
- Det er to definisjonen av eksentrisiteter, som egentlig er de samme, bare med ulikt fortegn. Det bør potensielt endres.
- Det er mulig at elementer ligger i retning motsatt av sin host *member*.
- Det er mulig å ha eksentrisitet i begge ender av en *member*, noe som fører til at ett element skal ha en eksentrisitet i en ende, og et annet element skal ha den i andre enden.
- Det er mulig å ha eksentrisiteter i begge endene av et element. Dette er unikt for bar *members*, som kun blir tildelt ett element. Andre typer *members* har minst to elementer.

**Merk:** Koordinater burde være definert inkludert eksentrisiteter, men dette er en svært omfattende endring som det ble valgt å sette på vent.

Beregning av elementkoordinater for Element-objekter ble plassert i egne *properties*, `ElementStartPoint` and `ElementEndPoint`. Disse tar hensyn til eksentrisiteter dersom de eksisterer.



Figur A.79: Modell feilen ble oppdaget i.



Figur A.80: Momentdiagram for modellen der feilen ble oppdaget før og etter.

#### A.4.6 Nullsetting av eksentrisiteter

ID	112
Klassifisering	Feil
Modifisert dato	20.05
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	15.05
Rapportert av	Espen



## Feil

Da eksentrisiteter av typen *MoveJoint* ble slettet ved å sette dem til 0 ble ikke *jointet* flyttet tilbake til sin opprinnelige posisjon, men den tilhørende *member* ble flyttet til *jointet*.

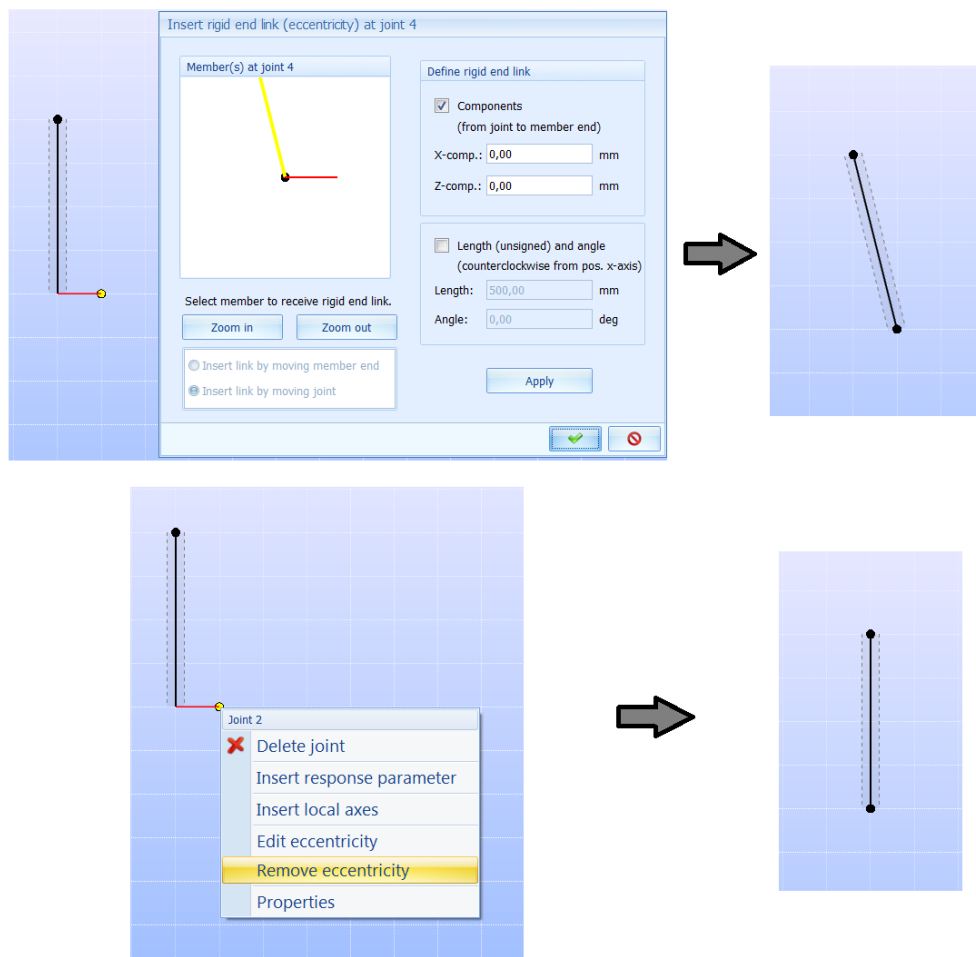
Å fjerne eksentrisiteten ved å klikke på valget "Remove eccentricity" fra høyreklikk-menyen fungerte.

## Endringer

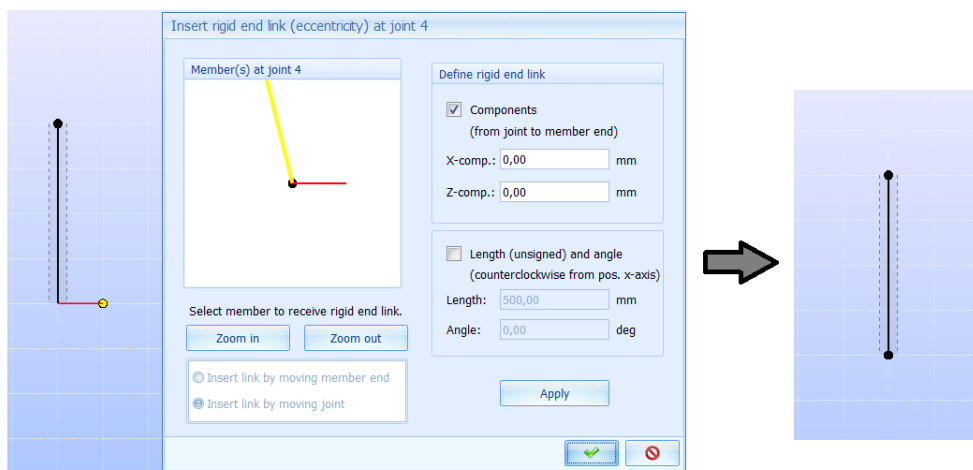
Årsaken var at det var glemt å sette *joint*-koordinater til opprinnelige koordinater.

**Merk:** endringene ble påvirket av omdefineringen av eksentrisiteter i A.4.9 Eksentrisiteter ved flere *MoveJoints*.

Namespace	Endrede klasser
fap2D.Gui	FrmEccentricityDialog.cs
fap2D.StructuralData	Eccentricity.cs



Figur A.81: Før - Sletting av eksentrisitet ved nullsetting flytter ikke *jointet* tilbake – "Remove eccentricity" i høyreklikk-menyen fjerner eksentrisiteten korrekt.



Figur A.82: Etter - Sletting av eksentrisitet ved nullsetting flytter *jointet* tilbake.

#### A.4.7 Validering av *arch members* for eksentrisiteter

ID	128
Klassifisering	Feil
Modifisert dato	16.05
Modifisert av	Espen
Rapportert i versjon	3.3.0 - DE
Rapportert dato	15.05
Rapportert av	Espen

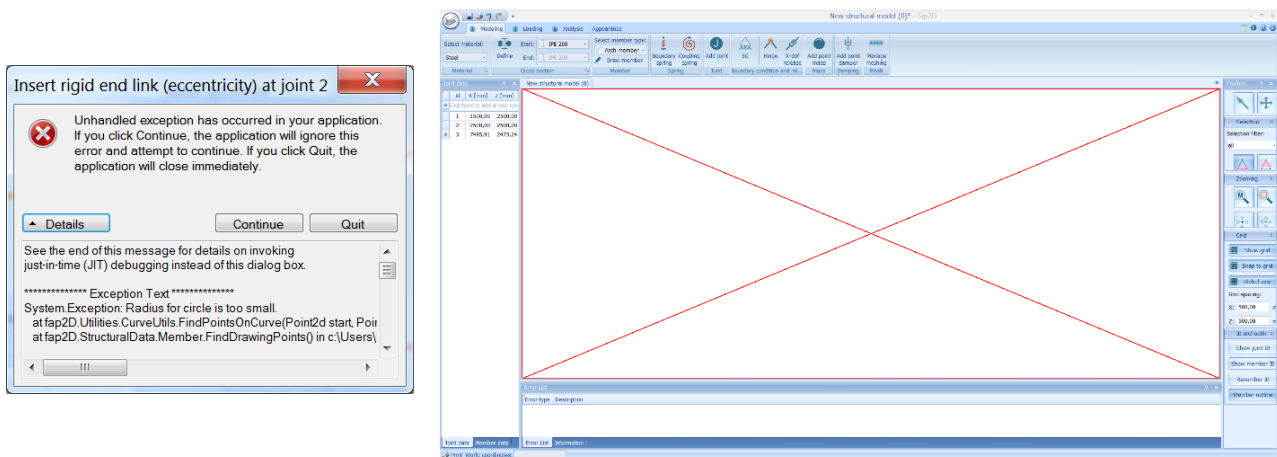
#### Feil

Det oppstod *unhandled exception* og fullstendig programstopp dersom eksentrisitet endret *circular arch member* for mye. Det var altså ikke noen sjekk for om ny *member*-geometri var gyldig før eksentrisitet godtas.

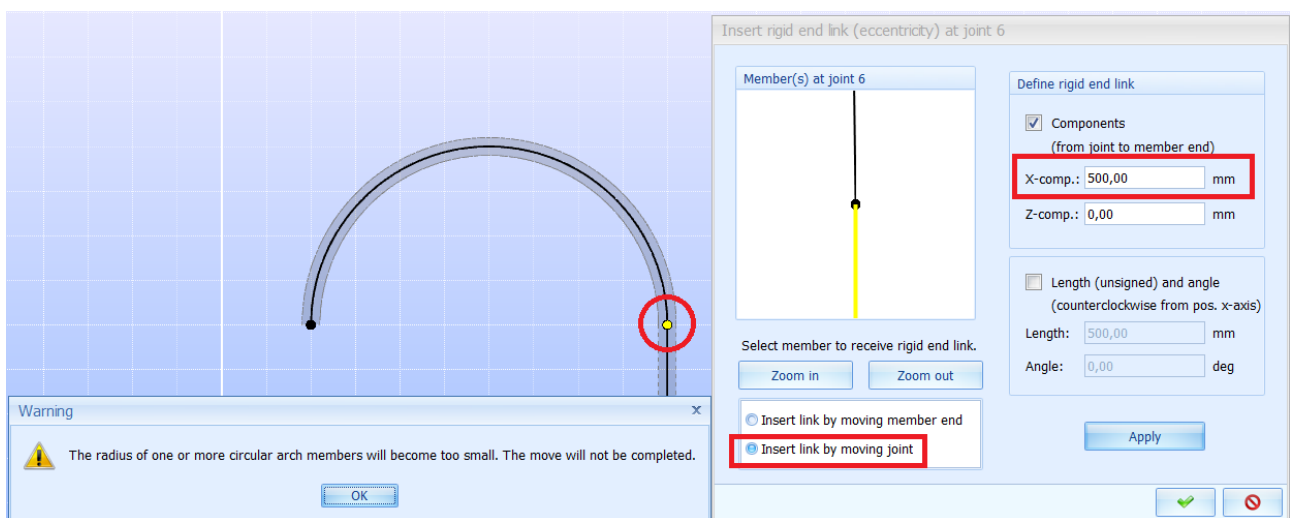
#### Endringer

Metoder som påfører endringer går nå gjennom hver tilknyttede *member* for å sjekke om de har gyldig geometri etter at eksentrisiteten er lagt til.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmEccentricityDialog.cs UC_Eccentricities.cs
fap2D.StructuralData	Eccentricity.cs



Figur A.83: Før – *unhandled exception* når *arch member* blir ugyldig.



Figur A.84: Etter - Advarsel vises hvis *arch member* blir ugyldig.

## Detaljer

Metodene som sjekker om geometrien er gyldig tar utgangspunkt i, og benytter noen av, metodene nevnt i feil A.4.1 Validering av *arch member*-geometri.

Det sjekkes både om *arch memberen* som flyttes, og om *arch members* som er tilknyttet *jointet* som flyttes, har gyldig geometri

### A.4.8 Tegning av *MoveJoint*

ID	129
Klassifisering	Feil
Modifisert dato	16.05
Modifisert av	Espen
Rapportert i versjon	3.3.0 - DE
Rapportert dato	15.05
Rapportert av	Espen

## Feil

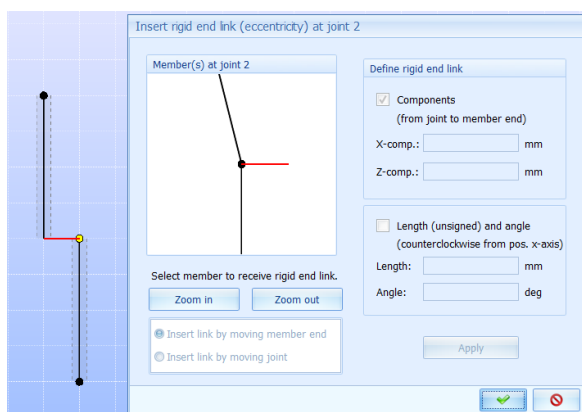
Eksentrisiteter av typen *MoveJoint*<sup>3</sup> ble tegnet feil i *user control* (UC\_Eccentricity.cs).

Metoden `GenerateSubMemberLines()` og metodene for *joint*-koordinater tok ikke hensyn til at et *joint* var flyttet, og tegnet *submembers* fra posisjonen til *jointet* uten eksentrisitet.

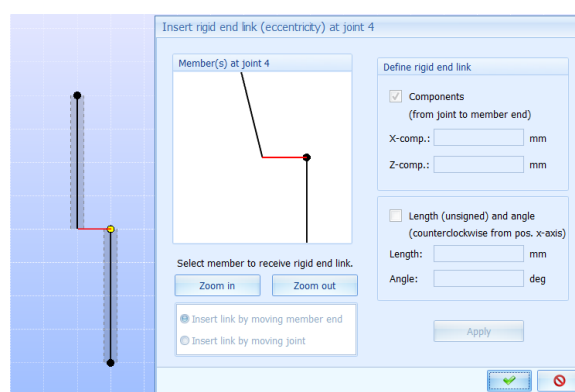
## Endringer

Metoden `GenerateSubMemberLines()` ble endret for å bruke *joint*-koordinater som inkluderte eksentrisitet.

Namespace	Endrede klasser
fap2D.Gui	UC_Eccentricities.cs
fap2D.StructuralData	Joint.cs



Figur A.85: Før – *joint* er flyttet vekk fra øverste *member*, men dette vises ikke.



Figur A.86: Etter – *joint* er flyttet og reflekteres i figuren.

### A.4.9 Eksentrisiteter ved flere *MoveJoints*

ID	130
Klassifisering	Feil
Modifisert dato	21.05
Modifisert av	Espen
Rapportert i versjon	3.3.0 - DE
Rapportert dato	15.05
Rapportert av	Espen

## Feil

Flere eksentrisiteter av typen *MoveJoint* på samme *joint* gir tegnefeil og merkelig oppførsel. To stykker *MoveJoint* eller først *MoveJoint* så *MoveMember* gir feil som *exception* og programstans.

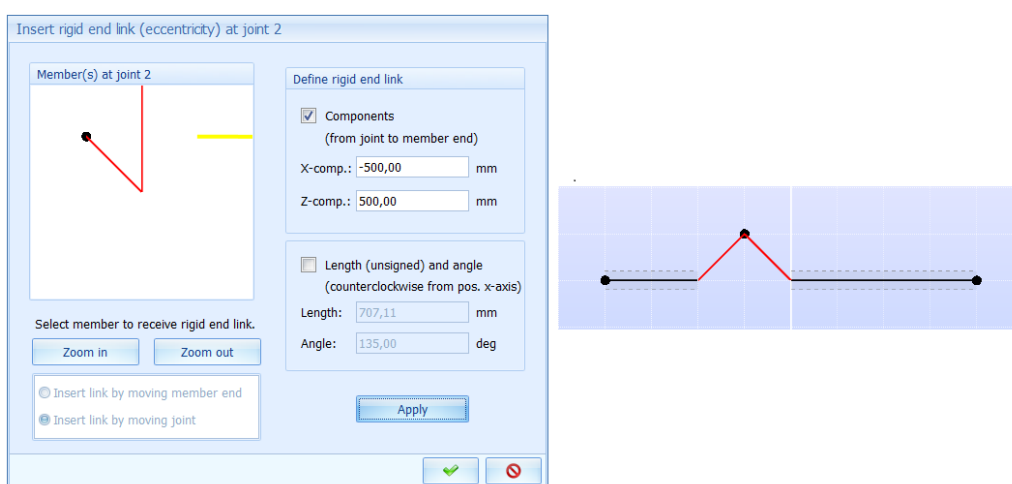
<sup>3</sup> en eksentrisitet som flytter *jointet* i forhold til en valgt *member* ende

## Endringer

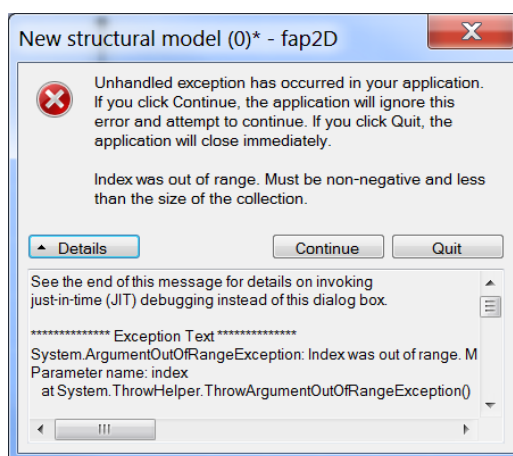
Definisjonen av eksentrisitetstypen *MoveJoint*, som flytter *joint*, er endret slik at eksentrisiteten gir *jointet* en helt ny posisjon, men original posisjon kan beregnes på grunnlag av dets eksentrisiteter.

Det er også endret slik at *joints* med flere *MoveJoint*-eksentrisiteter ikke krasjer programmet, og figuren i eksentrisitets-dialogen oppdaterer seg korrekt.

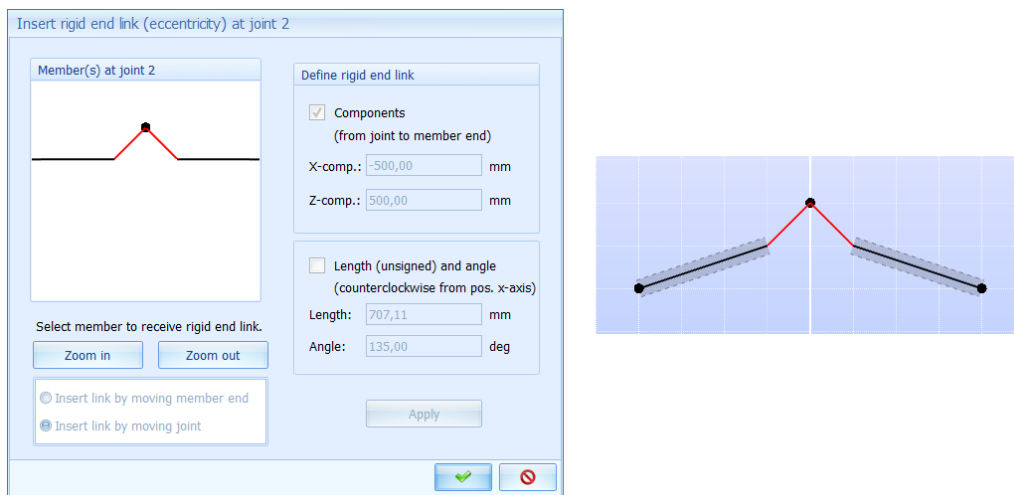
Namespace	Endrede klasser
fap2D.Gui	FrmEccentricityDialog.cs UC_Eccentricities.cs
fap2D.StructuralData	Eccentricity.cs Joint.cs



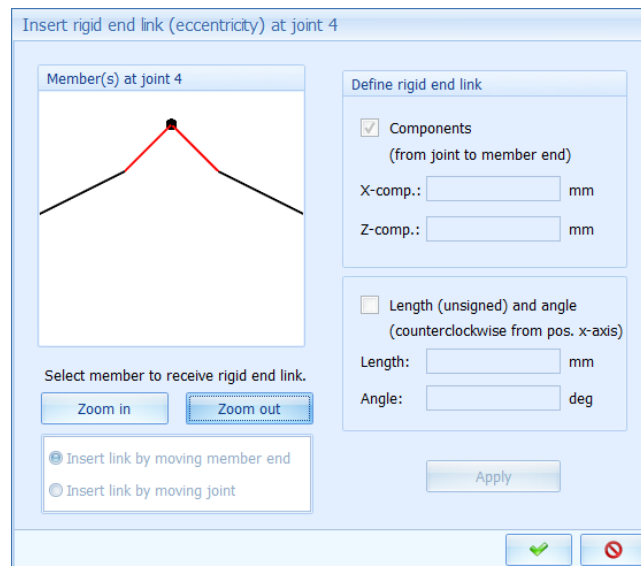
Figur A.87: Før – t.v. Eksentrisitet ved å flytte *jointet* som kobler to horisontale *members* - venstre *member* flytter *jointet* (500 mm, 500 mm), høyre *member* flytter *jointet* (-500 mm, 500 mm) – t.h. Modellen etter å ha satt inn eksentrisiteter - *jointet* skulle være flyttet (500 mm - 500 mm, 500 mm + 500 mm), men er kun flyttet avstanden som var satt på sist.



Figur A.88: Før - Å redigere eksentrisiteten i figur A.87 er umulig siden tegnemetoden krasjer når man prøver å åpne dialogen.



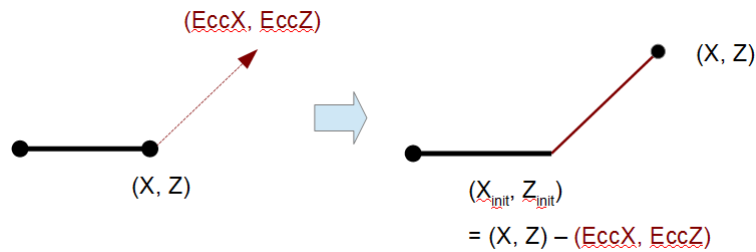
Figur A.89: Etter – t.v Samme eksentrisitet som i figur A.87 der tegnetoden viser endringene – t.h. Modellen tegnet korrekt.



Figur A.90: Etter – Figuren i eksentrisitets-dialogen tegnes korrekt.

## Detaljer

Implementasjonen av *MoveJoint* har vært feil, og antatt at et *joint* kun kan ha én eksentrisitet som flytter *jointet*. Siden *joint*-koordinatene er såpass grunnleggende i programmet er det gjort slik at eksentrisitet flytter *jointet*, men original posisjon kan beregnes. Dette er i motsetning til å regne ut faktisk *joint*-posisjon ut ifra original posisjon og eksentrisiteter.

Figur A.91: Definisjon av *joint* koordinater med eksentrisiteter.

Tegnetometoden ble endret i A.4.8 *Tegning av MoveJoint*. Den fungerer nå mye bedre, og er tilpasset flere *MoveJoint* uten å krasje programmet. Det bør nevnes at noe justering må til for at visningen skal korrekt reflektere endringen av *member* geometrien. Dette vil ikke være stort problem fordi *MoveJoint* er vanskeligere å bruke. *MoveJoint* eksentrisitet er mindre intuitiv enn *MoveMember* fordi *joints* kan ha flere eksentrisiteter som flytter på det.

#### A.4.10 Overlappende *joints*

ID	38
Klassifisering	Feil
Modifisert dato	20.03
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	
Rapportert av	Espen

#### Feil

Det er mulig å lage overlappende *joints*, dvs. to eller flere *joints* som har like koordinater. Overlappende *joints* kan oppstå dersom man oppretter nye *joints* i *joint data*-tabellen i *left panel*, eller ved å flytte et eksisterende joint vha.:

- Musepekeren
- *joint data*-tabellen
- *joint properties*-dialogen

Overlappende *joints* markeres per nå med rød farge.

#### Endringer 3.3

Det er ikke lenger mulig å skape overlappende *joints*. Et *joint* som flyttes til en posisjon der det eksisterer ett fra før vil bli slettet sammen med alle egenskapene *joint*-et har (opplagerbetingelser, responsparametere o.l.).

Dersom det flyttende *joint*-et er knyttet til en eller flere *members* vil *member*-ene slettes dersom feil 5 inntreffer (*member* med overlappende *joints*). Dersom feil 5 ikke inntreffer skal alle tilknyttede *members* knyttes til *joint*-et som allerede eksisterer i den gitte posisjonen.

Hovedimplementasjonen som er gjort for å rette feilen er den nye metoden `DeleteDuplicateJointAndConnectedMembers()` i klassen `Joint`. Metoden tar også hånd om feil 5.

**NB: Det er fortsatt mulig å lage overlappende *members* i programmet! Se figur A.97 for eksempel.**

### Detaljer 3.3

Vi har brukt mye tid på å rette opp denne feilen. Ikke bare fordi vi mente den var høyt prioritert, men fordi rettingen omfatter de fleste fremgangsmåtene for å lage og endre konstruksjoner i **fap2D** (og inneholdt derfor mye nyttig lærdom).

Ved å begynne med denne feilen tidlig lærte vi raskere kildekoden å kjenne og kunne overføre kunnskapen vår til lignende oppgaver senere i masteroppgaven. I tillegg har vi fått dokumentert hvordan fremtidige utviklere kan angripe lignende problemer senere. Kunnskapsoverføring og dokumentasjon har nemlig vært en av våre tilleggsoppgaver i masteroppgaven.

### Endringer

Det er lagt til en innstilling i settings-dialogen som gir brukeren mulighet til å slette det flyttende joint-et automatisk når det overlapper med et annet joint.

Dersom denne innstillingen ikke er i bruk vil brukeren alltid få opp en dialog med valg om å enten slette det flyttende joint eller å beholde begge joints selv om de har samme koordinater.

Dersom man velger å beholde overlappende joints vil de heller ikke bli slettet fra modellen selv om de senere i modelleringen overlappes med andre joints. Overlappende joints vil markeres med rødfarge som før.”

### Detaljer/Begrunnelse

Vi fant ut at endringene som ble gjort i versjon 3.3.0 var ikke optimale. Veileder påpekte at det i noen tilfeller kan være behov for å legge to eller flere joints oppå hverandre. Dette vil nemlig skape muligheter for å lage quasi-3D-modeller / 2.5D-modeller.

Den nye innstillingen gjør programmet mer fleksibelt samtidig som brukeren har god kontroll på modelleringen.

<i>Namespace</i>	Endrede klasser
<code>fap2D.StructuralData</code>	<code>Joint</code>
<code>fap2D.Gui</code>	<code>FrmMainGui.FrmModel</code>
	<code>FrmMainGui.LeftPanel</code>
	<code>FrmJointPropertiesDialog</code>



1	0,00	0,00	■ 1	■ 2	1	0,00	0,00	■ 2
2	500,00	0,00			2	0,00	0,00	

Figur A.92: Før endring.

1	0,00	0,00	■ 1	■ 2	1	0,00	0,00	■ 1
2	500,00	0,00						

⚠ You have created overlapping joints. Joint 2 has been deleted. 16:25:55

Figur A.93: Etter endring.

1	0,00	-500,00	■ 1	■ 2	0,00	-500,00	■ 2
2	500,00	-500,00			0,00	-500,00	

Figur A.94: Før endring. Se feil 5 for detaljer.

1	0,00	-500,00	■ 1	■ 2	1	0,00	-500,00	■ 1
2	500,00	-500,00						

⚠ You have created overlapping joints. Joint 2 has been deleted.  
 ⚠ You have created overlapping joints. Member 1 has been deleted.

Figur A.95: Etter endring. Se feil 5 for detaljer.

Id	X [mm]	Z [mm]
* Click here to add a new row		
1	2000,00	1500,00
2	2000,00	2000,00
> 3	3500,00	2000,00

Id	X [mm]	Z [mm]
* Click here to add a new row		
1	3500,00	2000,00
2	2000,00	2000,00
> 3	3500,00	2000,00

Figur A.96: Før endring. To overlappende *joints* med hver sin opplagerbetingelse.

Id	X [mm]	Z [mm]
* Click here to add a new row		
1	2000,00	1500,00
2	2000,00	2000,00
> 3	3500,00	2000,00

Id	X [mm]	Z [mm]
2	2000,00	2000,00
3	3500,00	2000,00

⚠ You have created overlapping joints. Joint 1 has been deleted.

Figur A.97: Etter endring. Flyttende *joint* slettes, eksisterende beholdes. De to *members* blir overlappende. Markerer med mørkere *member outline*.

#### A.4.11 Members med joints i samme punkt

ID	5
Klassifisering	Feil
Modifisert dato	20.03
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	September 2013
Rapportert av	Torjus

## Feil

Det er mulig å lage en *member* der *joint*-ene i enden får samme koordinater. Dvs. at *member*-en ikke får utstrekning.

Feilen oppstår når man flytter det ene *joint*-et, med musepekeren, slik at avstanden til det andre blir mindre enn en halv *grid spacing*.

Den oppstår også når man plasserer det ene oppå det andre ved å endre koordinatene via *joint data*-tabellen i *left panel* eller via *joint properties*-dialogen. Etter at *joint*-ene er flyttet er det umulig å markere *member*-en.

## Endringer - Versjon 3.3.0 Developer

Dersom et *joint* flyttes, med musepekeren, vil det flyttende *joint* og den tilhørende *member*-en slettes når avstanden mellom de to *joint*-ene blir for liten.

Ved sletting av *joints* og *members* blir brukeren varslet via *Error List* i *bottom panel*.

Feilen håndteres vha. den nye metoden `DeleteDuplicateJointAndConnectedMembers()`. Henger sammen med feil A.4.10, Overlappende *joints*.

Namespace	Endrede klasser
fap2D.StructuralData	Joint
fap2D.Gui	FrmModel FrmMainGui.LeftPanel FrmJointPropertiesDialog

1	1000,00	2000,00	
2	2000,00	2000,00	
1	1000,00	2000,00	
2	1000,00	2000,00	

Figur A.98: Før endring.

1	1000,00	2000,00	
2	2000,00	2000,00	
1	1000,00	2000,00	

Error List	
Error type	Description
	You have created overlapping joints. Joint 2 has been deleted.
	You have created overlapping joints. Member 1 has been deleted

Figur A.99: Etter endring. Brukeren varsles dersom *joints* og *members* slettes.

## Detaljer - Versjon 3.3.0 Developer

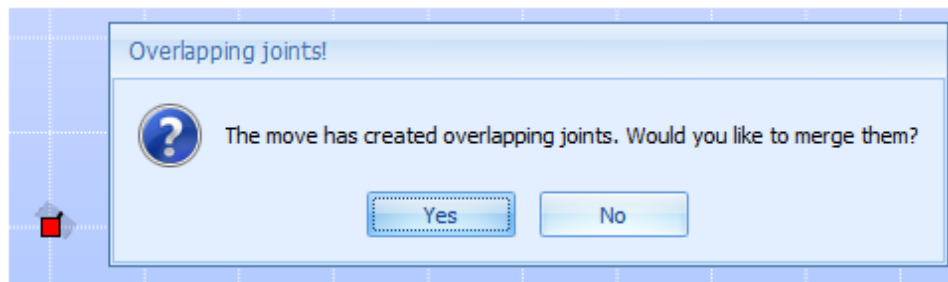
Grunnen til at feilen er problematisk er at den gir opphav til en *unhandled exception* som stopper programmet. Det er ikke problematisk å prøve å analysere modeller med slike *members*, for det vil bare føre til at analysene blir avbrutt og brukeren blir varslet.”

Det ble vurdert å lage en løsning som sørget for at *joints* som ble flyttet oppå andre *joints* automatisk ble flyttet tilbake til sin forrige posisjon. Grunnen til at det ikke er gjort er at man kan flytte dem tilbake ved å bruke *undo*-funksjonaliteten.

### Endringer - Versjon 3.1

Det er lagt til en innstilling i *settings*-dialogen som gir brukeren mulighet til å slette det flyttende *joint*-et automatisk når det overlapper med et annet *joint*. Se punkt A.9.9, *Merge overlapping joints setting*, for mer informasjon.

Dersom denne innstillingen ikke er i bruk vil brukeren alltid få opp en dialog med valg om å enten slette det flyttende *joint* eller å beholde begge *joints* selv om de har samme koordinater.



Figur A.100: Etter endring. Brukeren velger mellom å slå sammen *joints*, ved å slette det flyttende, eller å beholde begge *joints*.

Dersom man velger å beholde overlappende *joints* vil de heller ikke bli slettet fra modellen selv om de senere i modelleringen overlappes med andre *joints*.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmModel FrmMainGui.LeftPanel FrmJointPropertiesDialog

### Detaljer - Versjon 3.1

Endringene som ble gjort i versjon **3.3.0 Developer** passer godt til å løse feil A.4.11, men mindre godt til å løse feil A.4.10. Det er derfor er den nye innstillingen er lagt til.

Dersom innstillingen ikke er i bruk og man velger å beholde begge *joints* vil feil A.4.11 kunne oppstå, men brukeren har mulighet til å rette opp feilen ved å benytte *undo*-funksjonen. Feilen er heller ikke spesielt alvorlig siden den sjelden vil oppstå. Når den oppstår fører den ikke til noen *exception* lenger, så vi anser endringene som mer enn tilfredsstillende. Se A.4.10 for mer informasjon.

## A.4.12 Endringer i *member properties*

### Endring

Dialogboksen for *member properties* (egenskaper for *members*) fulgte ikke alltid dokumentasjonen, spesielt med hensyn til tverrsnittstyper. I tillegg ble enkelte egenskaper vist på en forvirrende måte.

Dialogboksen er endret til å vise teksten "<multiple>"<sup>a</sup> dersom det er flere typer av et objekt.

*Bar/cable/strut members* har ikke lenger muligheten til å ha varierende tverrsnitt.

Tverrsnitt kan kun interpoleres mellom samme type parametriske tverrsnitt. Interpolerte tverrsnitt markeres med teksten "<interpolated>".

<sup>a</sup>"<multiple> ble valgt istedenfor et mer spesifikt alternativ fordi det tar liten plass og er forståelig i kontekst av beskrivende *label*

### Metadata

ID:	113
Klassifisering:	Endring
Sist modifisert:	08.05
Modifisert av:	Espen
Rapportert i versjon:	3.0
Rapportert dato:	april 2014
Rapportert av:	Espen

### Endrede klasser

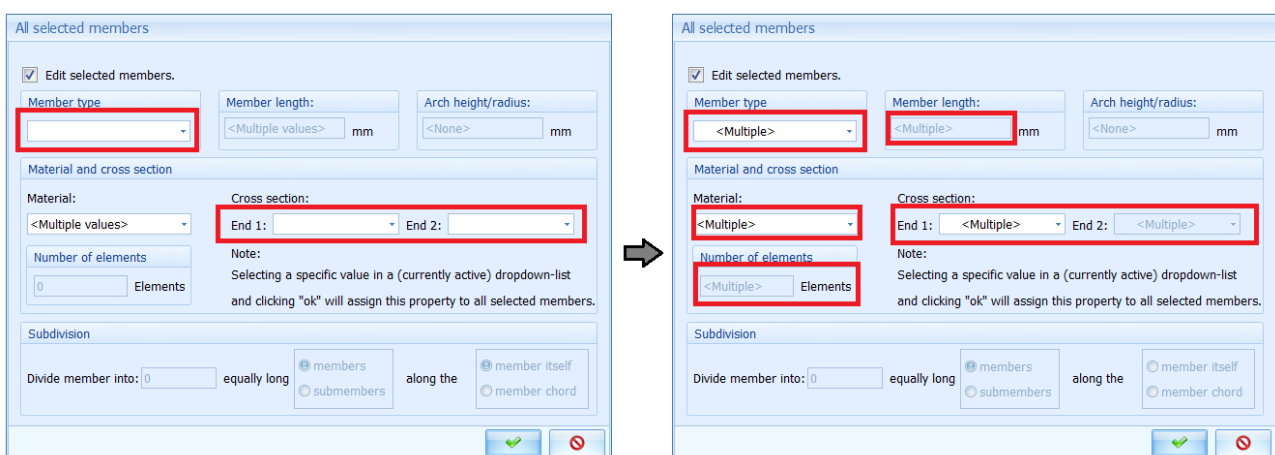
fap2D.Gui	FrmMember- Properties.cs
fap2D- .StructuralData	Member.cs SubMember.cs

### Detaljer

#### Visning av flere typer *members*

Før ble det ikke satt inn en plassholder i nedtrekkslistene dersom det var flere typer av et objekt. Programmet fungerte greit og nedtrekkslistene var fylt med verdier med alternative valg, derfor var det små endringer som var nødvendig for å fikse feilen.

Dialogboksen er endret til å vise "<multiple> dersom det er flere typer av et objekt i samlingen. Dette valget fjernes dersom man velger ett av alternativene, og alle *members* eller *submembers* blir overskrevet med den nye verdien.

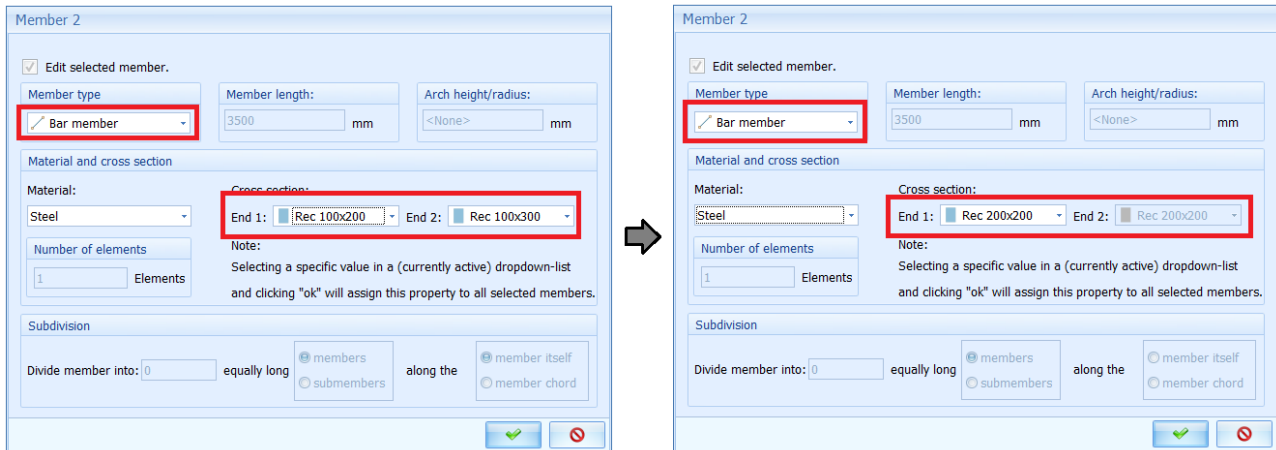


Figur A.101: Før - blanke felter for ulike verdier; Etter - Teksten "<Multiple> forteller bruker at de valgte *members* har ulike verdier.

#### Variierende tverrsnitt for *bar members*

*Bar/cable/strut members* skal ikke ha muligheten til å ha varierende tverrsnitt fordi de kun blir tildelt ett element. Elementer har konstant tverrsnitt, derfor er det misvisende å påstå at varierende tverrsnitt blir tatt til betraktning. Det vil også være uklart hvilke tverrsnittsparemetere som ville blitt brukt i beregningene.

**Merk:** Man kan lage en *bar member* (ikke *cable* eller *strut*) med varierende tverrsnitt ved å modellere en *beam member* med ledd i hver ende

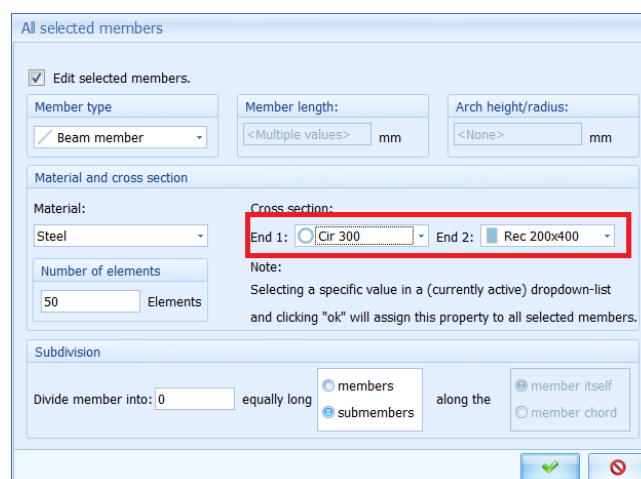


Figur A.102: Før - *bar member* kan ha varierende tverrsnitt – her: Rec 100x200 og Rec 100x300; Etter - *bar members* kan kun ha konstant tverrsnitt.

### Interpolering av parametriske tverrsnitt

I dokumentasjonen står det at parametriske tverrsnitt kan interpoleres, men bare så lenge tverrsnittene er av samme type parametriske tverrsnitt, e.g. rektangulært eller sirkulært. Det var før mulig å interpolere på tvers av parametriske tverrsnittstyper

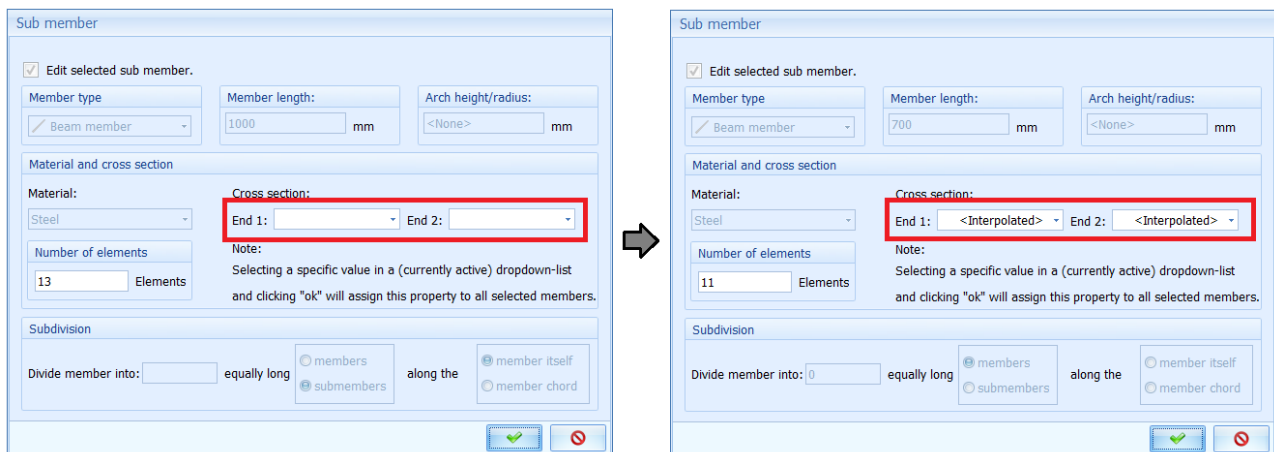
I praksis håndterte programmet dette tilfellet, men hvordan den håndterte det var avhengig av hva brukeren hadde gjort. Brukeren vil da ikke være sikker på hva som regnes på.



Figur A.103: Før - Det var mulig å interpolere tverrsnitt med ulike parametriske typer.

**Interpolert tverrsnitt** Tverrsnitt som var interpolert ble ikke vist i *member properties*, men ble vist med et blankt felt.

Dette er endret til å vises på samme måte som med flere ulike tverrsnitt, men istedenfor å vise "<multiple>" står det "<interpolated>".



Figur A.104: Før - blankt felt for å vise at et tverrsnitt var interpolert; Etter - teksten "<interpolated>" viser at tverrsnittet er interpolert.

### A.4.13 Sletting av overlappende *members*

#### Feil

Dersom man forsøkte å slette flere *joints* og *members* i en konstruksjonsmodell der de overlapper kunne det oppstå feilmelding og programstans. Feilen kunne også opptre i sammenheng med komplekse modeller med mange *joints* og *members* koblet til hverandre, selv om det ikke var overlapp.

#### Endring

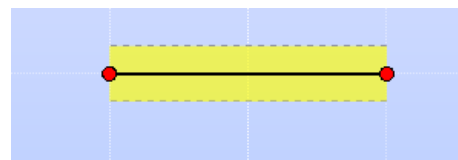
Løsningen var å la programmet ignorere feilmeldingen fordi dataobjektene som ikke ble funnet skulle fjernes av metodekallet. Altså var arbeidet allerede gjort.

#### Metadata

ID:	26
Klassifisering:	Feil
Sist modifisert:	24.05
Modifisert av:	Espen
Rapportert i versjon:	3.0
Rapportert dato:	Høst 2013
Rapportert av:	Espen

#### Endrede klasser

fap2D.StructuralData Member.cs



Figur A.105: Eksempelmodell med flere overlappende *members* og *joints* som kunne føre til feilen.

#### Detaljer

Feilmeldingen som ble vist sa:

System.ArgumentException: Cannot remove the specified item because it was not found in the specified Collection. at System.Collections.CollectionBase.System.Collections.IList.Remove(value) ...

Årsaken var at det oppstod null-referanser mellom *members* og *joints*. Hvilken rekkefølge *structural items* slettes i er noe uoversiktlig i programmet, slik at referanser til *joints* fra *members* pekte til *joints* som allerede var slettet.

#### A.4.14 Member-retning

ID	85
Klassifisering	Endring
Modifisert dato	19.05
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	07.05
Rapportert av	

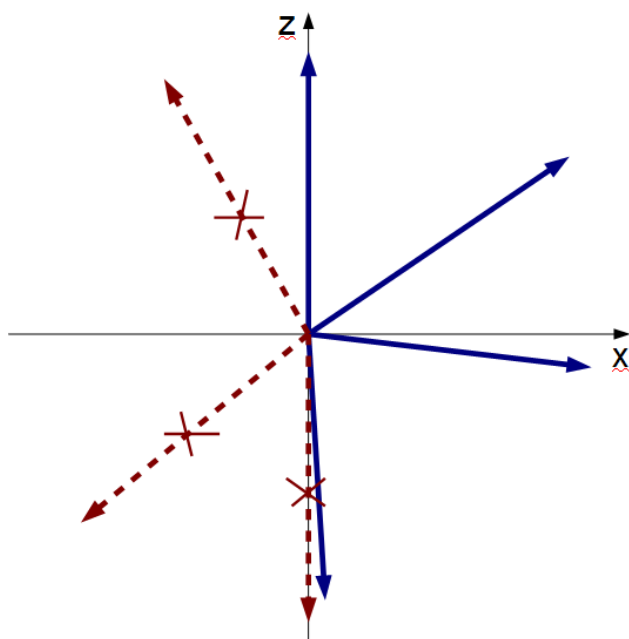
#### Endringer

Endrer retning av *members* slik at alle følger reglene om venstre mot høyre, i motsetning til å kunne være orientert i tilfeldig retning.

For å unngå å begrense måten brukere modellerer er det bestemt at *members* kan utplasseres med vilkårlig retning. Programmet endrer retning på *members* etter at bruker har valgt å kjøre analyse, før *meshing* blir gjennomført. På den måten vil ikke bruker merke noen endringer mens modellering foregår.

Betingelsen for at en *member* bytter retning er at en *member* med *joint* i ende 2 er til venstre for *joint* i ende 1, dersom begge *joints* er på samme x-koordinat må en *member* bytte retning hvis *jointet* i ende 2 er under *jointet* i ende 1. Se figur A.106.

<i>Namespace</i>	Endrede klasser
fap2D.StructuralData	Member.cs
fap2D.Computations	ComputationalModel.ModelTools.cs



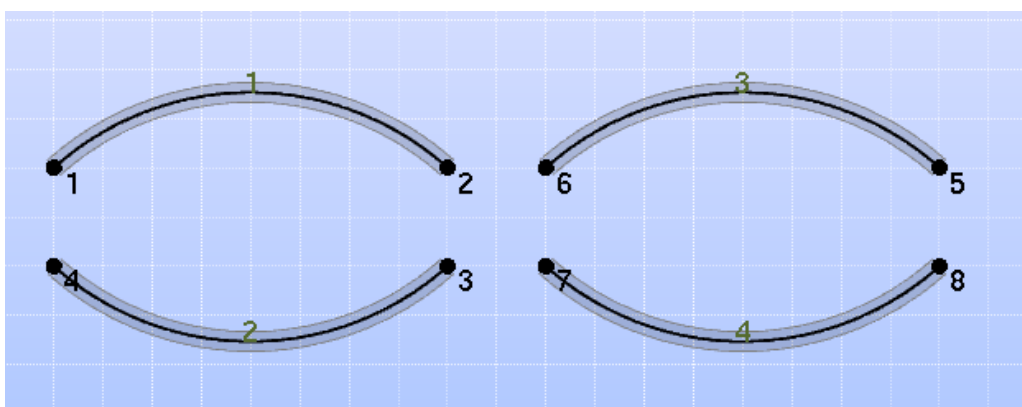
Figur A.106: *Members* med ende 1 i origo - *members* med stiplet linje må skifte retning.

### Detaljer

Opprinnelig har det vært tenkt at *members* skal være definert fra venstre mot høyre, evt. nedenfra og opp når en *member* er vertikal. Likevel har *members* frem til nå kunne ha tilfeldig retning avhengig av hvordan bruker setter dem ut.

Et problem som oppstår når en *member* har tilfeldig orientering er at modellering og resultater vises på en forvirrende måte. Dette poengteres i de følgende eksemplene.

**Eksempel:** I modellering blir høyde/radius og rekkefølgen på tverrsnitt påvirket av retningen til *members*. Som man kan se i figur A.107 er det intuitivt å anta at de øverste *arch members* (1 og 3) har positiv radius<sup>4</sup> fordi de krummer opp, mens *arch member* 2 og 4 har negativ radius. I realiteten har begge de venstre *arch members* (1 og 2) positiv radius. Dette er mulig ved at det lokale aksesystemet er rotert for *arch member* 2 og 3, noe man kan se på rekkefølgen *joints* er satt ut.

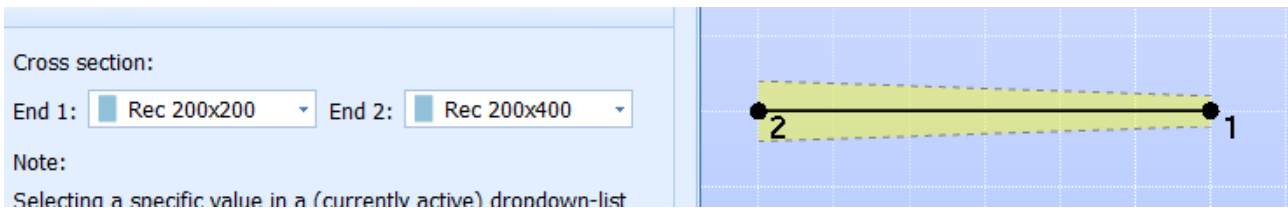


Figur A.107: *Arch member* 1 og 2 med positiv radius (3000 mm), *arch member* 3 og 4 med negativ radius (-3000 mm).

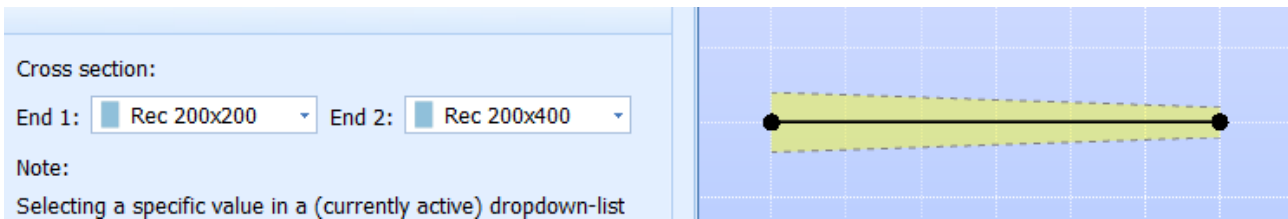
<sup>4</sup>Denne oppførselen gjelder også paraboliske *arch members* med høydeparameter



**Eksempel:** Figur A.108 viser en *member* med det lave tverrsnittet (*Rec 200x200*) i ende 1, og et høyt tverrsnitt (*Rec 200x400*) i ende 2. Dersom man fjerner *joint*-nummereringen, som i A.109, kan det virke som uoverensstemmelse mellom figur og dialog.



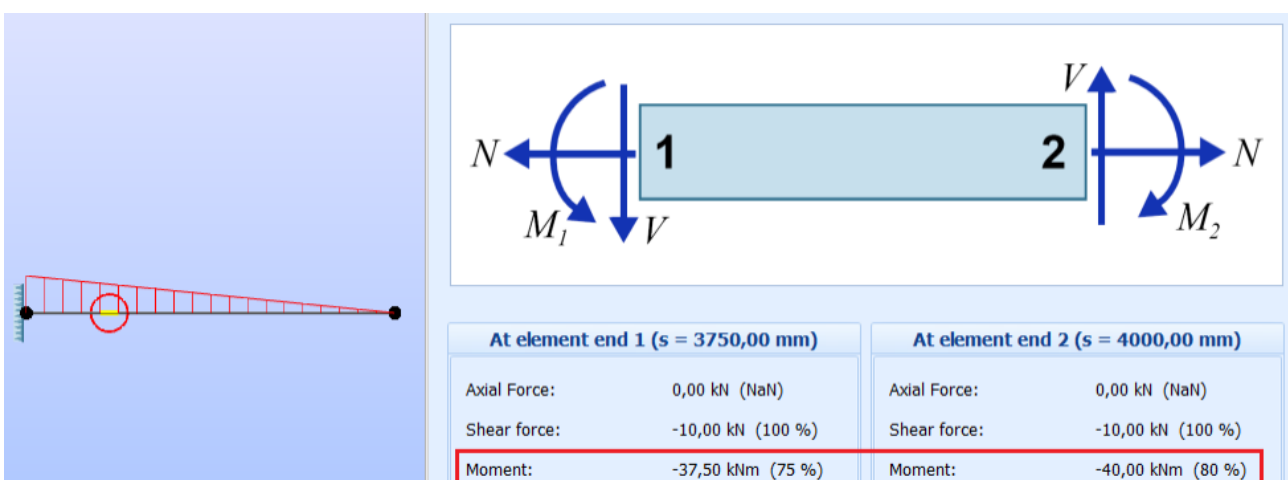
Figur A.108: *Member* med ulik tverrsnittstørrelse i hver ende - Ende 1 (*joint* 1) har *Rec 200x200*, ende 2 (*joint* 2) har *Rec 200x400*



Figur A.109: Samme *member* som i figur A.108 - Høyre ende har *Rec 200x200*, venstre ende har *Rec 200x400*.

**Eksempel:** Den lokale aksens til en *member* vil bestemme retningen til elementene den danner grunnlaget for. Beregningene vil gi korrekte resultater, og elementer vil få verdier tilsvarende deres orientering. Dette betyr at elementer tegnet fra høyre mot venstre vil være orientert omvendt av det som vises i dialogboksen for hvert element. I figur A.110 er det moment på undersiden av elementet ifølge dialogboksen, noe som tilsynelatende ikke reflekteres av diagrammet på modellen.

Det kan også bemerkes at elementenes neste naboelement vil ligge til venstre for elementet dersom *memberen* er tegnet fra høyre mot venstre.



Figur A.110: Utkrager med elementer orientert fra høyre mot venstre.

### Snu *member*

For å unngå å begrense måten brukere modellerer er det bestemt at *members* kan plasseres med vilkårlig retning. Programmet endrer retning på *members* etter at bruker har valgt å kjøre

analyse, før *meshing* blir gjennomført. På den måten vil ikke bruker merke noen endringer mens modellering foregår.

Det kan argumenteres for at *members* bør settes tilbake til original orientering etter *meshing*, men det er hensiktsmessig å beholde den mer intuitive retningen (venstre mot høyre) til hver *member* fordi brukeren sannsynligvis ikke vil ha noe konkret forhold til en *member* og dens parametere. Dersom det likevel skulle være tilfellet er det enkelt å oppdage endringen og forholde seg til dem. Det er bestemt at bruker ikke trenger å få beskjed om at endringen er gjort fordi geometrien til modellen er uendret.

Følgende endringer, som gjennomføres når en *member* endrer retning, kan observeres av bruker:

- høyden/radius av *arch members* får omvendt fortegn
- Tverrsnittene bytter side i dialogboksen for *member*-egenskaper

Endringer som er skjult når en *member* endrer retning er:

- *memberens* ender bytter *joints* (*joints* vil ikke flyttes)
- *memberens* ender bytter eksentrisiteter

*Submembers* endrer *joint* tilsvarende den for sine *host members*. Alle *internal joints* og tilordninger til dem forblir uendrede. Tverrsnitt byttes om slik at *submembers* i starten av en *member* får de speilede tverrsnittene til *submembers* i slutten av *memberen*. Bruker vil ikke kunne oppdage endringer i *submember*.

Betingelsen for at en *member* bytter retning er at en *member* med *joint* i ende 2 er til venstre for *joint* i ende 1, dersom begge *joints* er på samme x-koordinat må en *member* bytte retning hvis *jointet* i ende 2 er under *jointet* i ende 1. Se også figur A.106.

I koden like før *meshing*:

```
...
if (m.Joint2.X < m.Joint1.X
    || (m.Joint2.X == m.Joint1.X && m.Joint2.Z < m.Joint1.Z))
    m.ChangeDirection();
...
```

### A.4.15 Sletting av tverrsnitt som er i bruk

#### Feil

Det er mulig å slette tverrsnitt som er i bruk fra listen over tverrsnitt. Dvs. at dersom man går inn i *member properties dialog* for *members* som benytter de et slettet tverrsnitt, kan det se ut som et annet tverrsnitt er brukt.

#### Endring

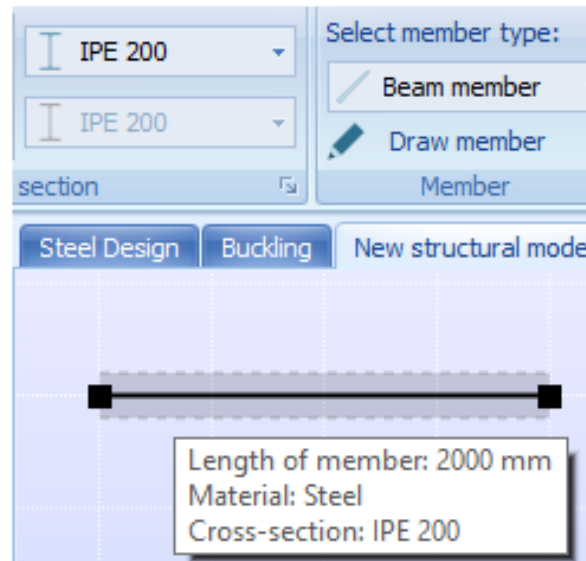
Det er ikke lenger mulig å slette tverrsnitt som er i bruk.

#### Argumentasjon

Det bør ikke gå an å slette tverrsnitt som er i bruk fra listen over tverrsnitt, fordi disse tverrsnittene vil fortsatt være knyttet til *members* som brukte dem før sletting fra listen.

Dessuten forvirres brukeren fordi det ser ut som et annet tverrsnitt er i bruk når man går inn i *member properties dialog*.

#### Figurer



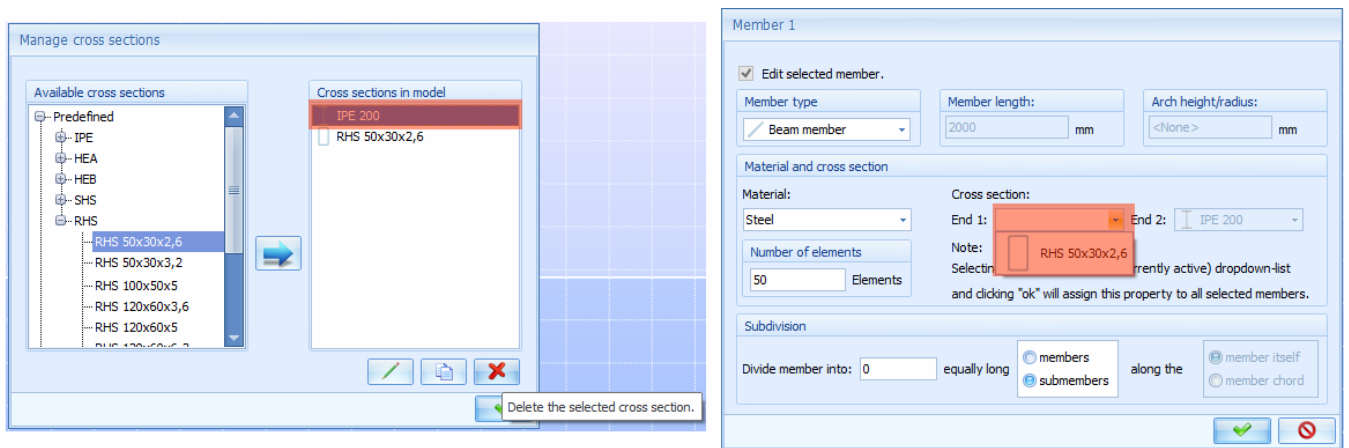
Figur A.111: *Member* med IPE 200-tverrsnitt.

#### Metadata

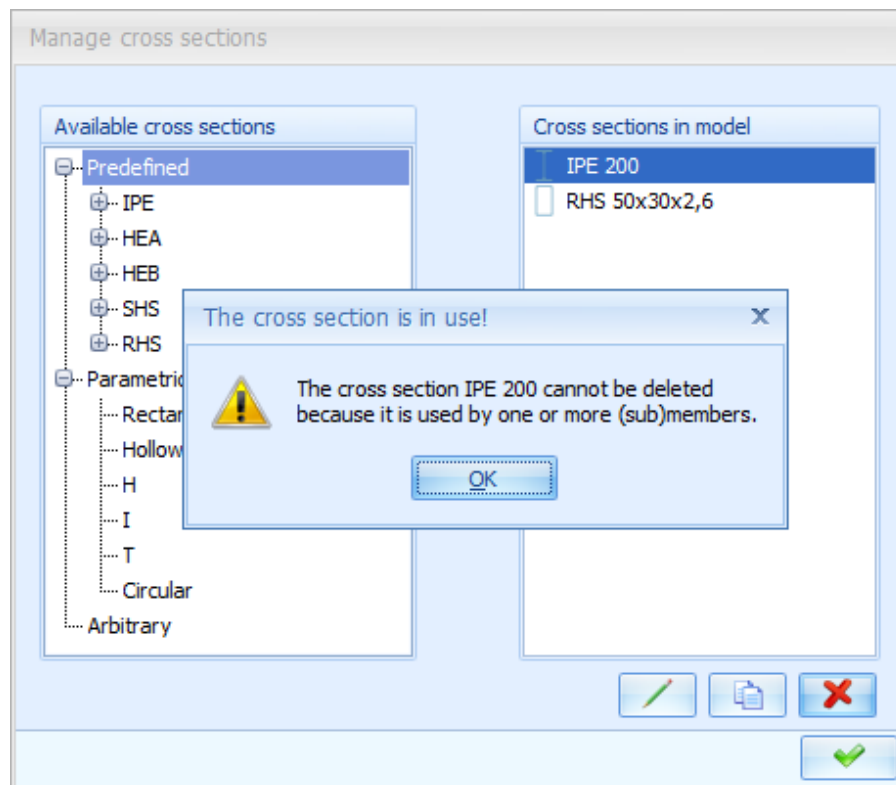
ID:	84
Klassifisering:	Feil
Sist modifisert:	13. mai
Modifisert av:	Espen, Torjus
Rapportert i versjon:	3.0
Rapportert dato:	?
Rapportert av:	Espen

#### Endrede klasser

fap2D.Gui FrmCrossSectionDialog



Figur A.112: Versjon 3.0: Tverrsnitt slettes (t.v). I *member properties dialog* (t.h) vil det være tvetydighet rundt hvilket tverrsnitt som er i bruk i *member*-en etter sletting.



Figur A.113: Versjon 3.1: Ikke lenger mulig å slette tverrsnitt som er i bruk.

### A.4.16 NullPointerException ved redigering av koblingsfjær

#### Feil

Fjær fikk `NullPointerException` når man skulle redigere fjæregenskapene. Feilen forekom fordi *submember*-ne som fjærer kobler sammen ble satt feil.

#### Endring

Endret skrivning til datastrukturen som inneholder referansene til *submember*-ne.

#### Metadata

ID:	125
Klassifisering:	Feil
Sist modifisert:	19.05
Modifisert av:	Espen
Rapportert i versjon:	3.0
Rapportert dato:	09.05
Rapportert av:	K. Bell

#### Endrede klasser

fap2D.Gui	UC_CouplingSpring.cs
fap2D.StructuralData	Spring.cs
fap2D.Storing	StoringUtilities.cs

#### Detaljer

Feilen var at en liste som holdt styr på hvilke *submembers* fjæren koblet sammen hadde feil referanser. Listen pekte da på null-felter, noe som gir feilmelding

En forenklet oversikt over hvordan feilen oppstod er vist nedenfor:

```
_subMember = new List<SubMember>();
...
_submember.Add(null);
_submember.Add(null);
...
... = _submember[0] ...
... = _submember[1] ...
```

### A.4.17 Feilaktig skalering av rotasjonsstivhet i *spring dialog*

ID	50
Klassifisering	Feil
Modifisert dato	25.02
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	21.02
Rapportert av	Petter H. Holmstrøm

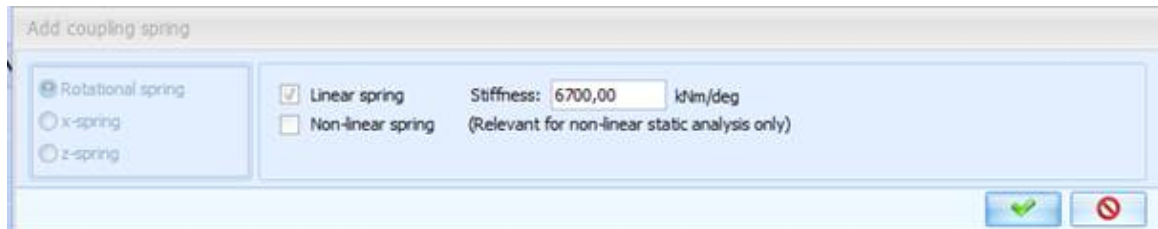
#### Feil

Verdien av rotasjonsstivheten til en rand- eller koblingsfjær ble endret hver gang man åpnet dialogboksen som viser dens egenskaper. Verdien ble endret med en faktor  $57.32 \approx \frac{180}{\pi}$ .

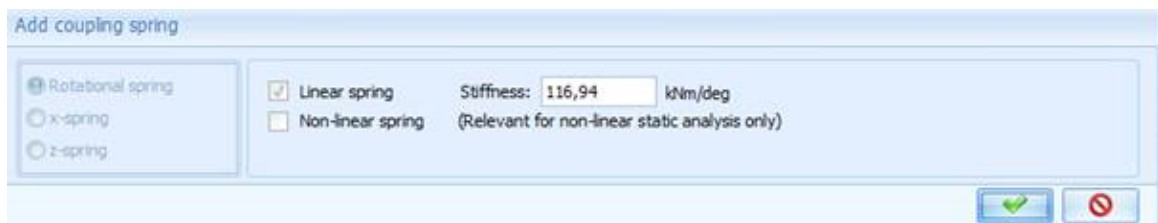
## Endringer

En *enum*-verdi i en metode som konverterer enheter (`Units.GetConversionFactor(...)`), var satt til feil verdi.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmSpringDialog



Figur A.114: Første gang man trykker *Edit Spring* er verdien korrekt.



Figur A.115: Andre gang man trykker *Edit Spring* er verdien endret.

## Detaljer

Når man redigerer rotasjonstivhet til en rand- eller koblingsfjær med lineær stivhet endres verdien uten at man selv ønsker det.

Endringen var alltid en faktor  $57.32 \approx \frac{180}{\pi}$ , noe som tydet på en omregning fra grader til radianer der det ikke var korrekt.

En *enum* som var parameter i en metode som henter faktor for konvertering mellom enheter var satt feil. Ved opprettelse av dialogboksen var *enum*'en satt til å konvertere fjærstivheten til riktig enhet for rotasjonstivhet. Ved lukking av dialogboksen var *enum*'en satt til å konvertere fjærstivheten til riktig enhet for aksialstivhet istedenfor rotasjonstivhet.

Følgende kodelinjer:

```
textEditStiffness.EditValue =
    _springInEdit.Stiffness[0] /
    Units.Instance.GetUnitConversionFactor(UnitType.axialStiffness);
```

Ble endret til:

```
textEditStiffness.EditValue =
    _springInEdit.Stiffness[0] /
    Units.Instance.GetUnitConversionFactor(UnitType.rotationalStiffness);
```

Dette var et tilfelle der man har kopiert en linje med kode som er nesten helt lik, men glemte å endre den viktigste detaljen. Mest sannsynlig skjedde det da man sentraliserte konvertering mellom enheter.

## A.5 Beregningsmodell og analyser

### A.5.1 Angi *load combination* for *buckling analysis*

#### Feil

Dersom brukeren ikke har angitt *load combination* (via *loading dialog*) for *buckling analysis*, og prøver å starte en analyse vil det vises en feilmelding i *error list*. Grunnen er at når det ikke er angitt *load combination* vil det ikke settes på last i analysen, og knekning vil ikke oppstå. **Frame2D** varsler derfor om at det er lave (ingen) trykkrefter:

“*Frame2D\_LB: Buckling analysis aborted due to very low levels of compression forces.*”

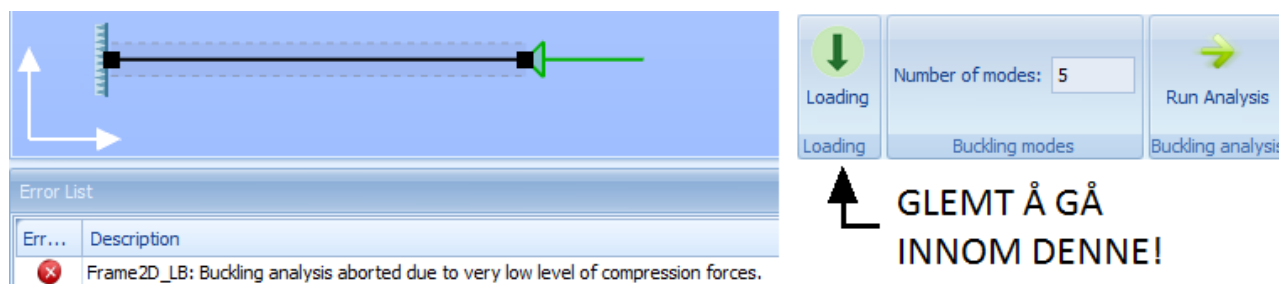
Det er vanskelig for brukeren å forstå at det ikke er angitt en *load combination*, for i versjon **3.0** er det mange av de andre analysene som har en *default load combination*.

#### Endring

Dersom brukeren ikke har angitt *load combination*, og prøver å starte en analyse, vil det vises en dialog som gjør brukeren oppmerksom på at *load combination* ikke er angitt. Brukeren blir så henvist direkte til *loading dialog* og kan angi en *load combination*. Deretter kan analysen startes.

Feilen har sammenheng med endringer gjort i punkt A.5.5, og er nevnt i avsnitt 3.4.3 i [10].

#### Figurer



Figur A.116: Versjon **3.0**: Brukeren har glemte å angi *load combination* og får en feilmelding som er vanskelig å tolke.

#### Argumentasjon

Feilen er vanskelig å tolke for brukeren fordi feilmeldingen ikke sier hva som må gjøres for å kunne kjøre analyse, den bare sier at det er for lite trykk i konstruksjonen.

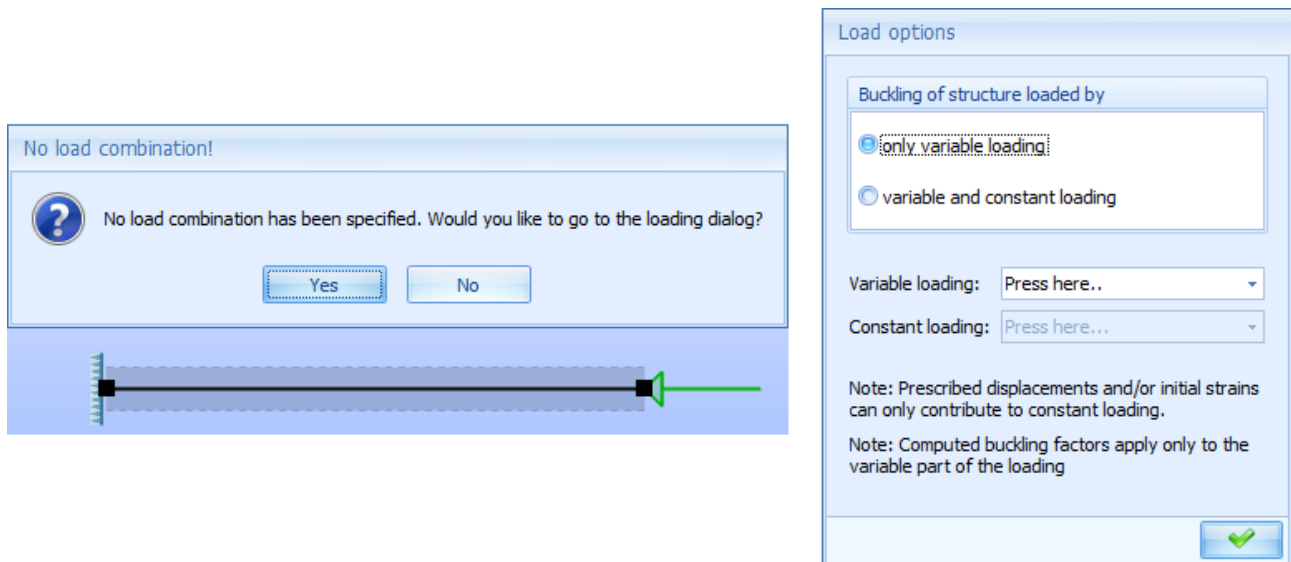
Flyten i programmet ødelegges og brukeren kan bli frustrert. Endringen gjør at flyten blir god og den lærer brukeren å alltid gå innom *loading dialog* før analysen startes.

#### Metadata

ID:	4
Klassifisering:	Feil
Sist modificert:	13. mai
Modifisert av:	Torjus
Rapportert i versjon:	3.0
Rapportert dato:	Høst 2013
Rapportert av:	Kolbein, Arne Aalberg

#### Endrede klasser

fap2D.Gui FrmMainGui.AnalysisPage



Figur A.117: Versjon 3.1: Brukeren har glemte å angi *load combination* og blir guidet til *loading dialog* for å gjøre det.

## A.5.2 Nedre grense for elementlengde

ID	35
Klassifisering	Feil
Modifisert dato	11.02
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	
Rapportert av	Espen

### Feil

Dersom man bruker for korte elementer i beregningsmodellen får man numeriske feil.

### Endringer

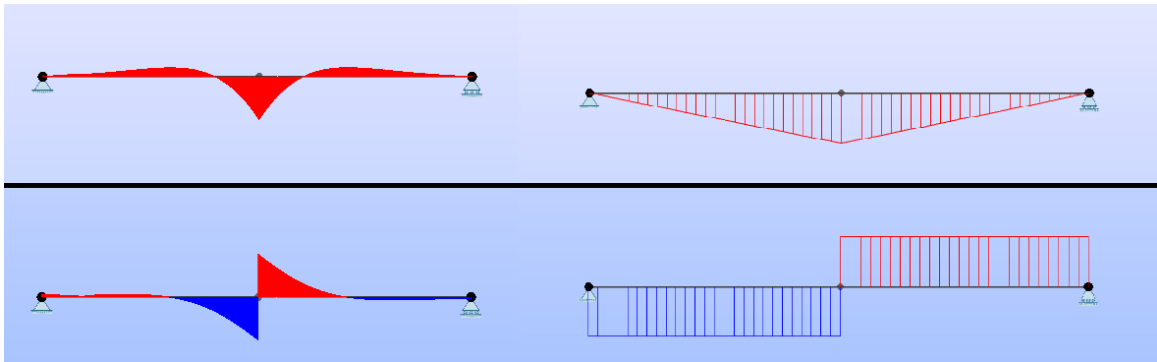
Programmet gir nå advarsler i form av dialogbokser dersom man har for korte elementer. Korte elementer er definert ved elementlengde kortere enn 5% av tverrsnitthøyden.

Advarslene kommer ved:

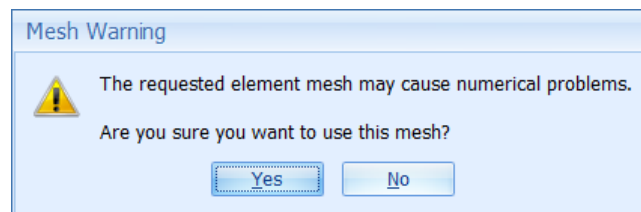
- redigering av *member properties*
- redigering av *mesh settings*
- kjøring av analyse



<i>Namespace</i>	Endrede klasser
fap2D.StructuralData	Member SubMember LoadPath
fap2D.Gui	FrmMainGui.AnalysisPage FrmMemberPropertiesDialog FrmManageMeshingDialog FrmModel
fap2D.Computations	ComputationalModel.LoadTrain ComputationalModel.ModelTools



Figur A.118: Før endring - Moment- og skjærtdiagram for korte elementer (venstre) og normale elementer



Figur A.119: Etter endring.

### A.5.3 Feil verdi i Mkey

ID	124
Klassifisering	Feil
Modifisert dato	28.04
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	21.03
Rapportert av	K. Bell

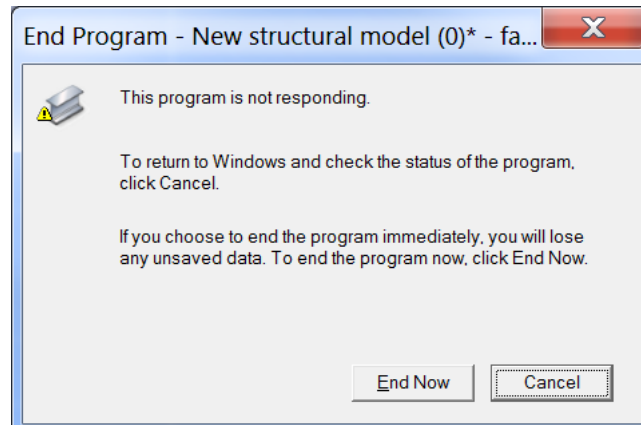
#### Feil

Ikkelineær statisk analyse fikk feil parametere da man regnet med kun foreskrevne forskyvninger (*prescribed displacements*).

## Endringer

Endret måten parametere ble valgt ut til Mkey slik at ekstern last ikke alltid må være tilstede for å kjøre analyse. Endingene ble gjort i metoden `RunNonLinearStaticAnalysis()`.

<i>Namespace</i>	Endrede klasser
<code>fap2D.Computations</code>	<code>ComputationalModel.Analysis.cs</code>



Figur A.120: Programmet sluttet å reagere og måtte avsluttes med Oppgavebehandler.

## Detaljer

Feilen forekom på to måter avhengig av input:

- Programmet hengte – kun konstant foreskreven forskyvning
- Melding "*too much loading*" – kun varierende foreskreven forskyvning [Merk: dette kan ha være på grunn av for få laststeg]

Feilen var at noen parametere som beskriver eksterne krefter, `Mkey(41)` og `Mkey(42)`, ble inkludert uansett om de eksisterte eller ikke. Istedenfor å bli satt til 0, som beskriver at de ikke er tilstede, ble de satt til enten 1 eller 2, som beskriver at de skal være inkludert som hhv. konstante eller variable.

Dette skjedde fordi det ble gjort antakelser om at det måtte være med ekstern last (annen last enn temperatur eller foreskreven forskyvning), enten som variabel, konstant eller begge typer last.

### A.5.4 Unøyaktig beregning av nodeplassering i *arch members*

Denne feilbeskrivelsen er komplisert og må lese i detalj for å kunne forstås!

#### Introduksjon

Plasseringen til nodene i *arch members* beregnes slik at buelengden mellom nodene er lik. Dvs. at de rette elementene som tegnes mellom nodene kan ha ulik lengde. Figur A.121 illustrerer dette (på en litt overdrevet måte).

I **fap2D** er det behov for å forholde seg til både buelengden mellom hver node, og den rette lengden (til elementene) mellom nodene, fordi:

- Buelengdene mellom elementene brukes til å regne ut hvor langt det er fra den første noden (i et elementet) tilbake til 1. joint (det venstre *joint* i figur A.121). Vi kaller denne verdien for “**s**”. Verdien vises i:
  - *Section force dialog*, dialog som viser krefter i elementene.
  - *Utilization dialog*, viser utnyttelse av kapasitet (i hht. ståldimensjonering), i elementene.
  - *Influence lines result dialog*, viser influenslinje-krefter i elementene.
  - Eksport av krefter i *members* (alle elementer/noder i *members*).
- Elementlengdene (rette) brukes derimot i alle beregningene som gjøres. Dvs. at resultatene som vises i punktene ovenfor er basert på bruk av rette elementer.

Grunnen til at buelengdene benyttes til å regne ut avstand fra 1. *joint* er at siste node i en *submember* skal ha en avstand som svarer til hele buelengden til *submember*-en. De andre nodene bør også presenteres riktig, men det er viktigst at avstanden til siste node er korrekt fordi den skal tilsvare den totale buelengden. Den totale buelengden vises nemlig når man holder over en *arch member*, så det er viktig at buelengden stemmer overens med avstanden til siste node (for konsistensens skyld).

**NB:** En viktig detalj som man må legge merke til er at det brukes en variabel for å lagre avstanden fra 1. *joint* til midtpunktet i et element, der midtpunktet er definert til å ligge langs buen! Dvs. at dette midtpunktet i realiteten ikke er midt på et element, det er projeksjonen fra elementets midtpunkt ut til buen. Dette har skapt mye forvirring for utviklerne. Det høyre bildet i figur A.121 illustrerer dette.

### Feil

Ved beregning av avstanden fra 1. *joint* i en *submember* og fram til nodene i *submember*-ens elementer er det gjort feil.

For *arch members* er det nemlig slik at avstanden fram til nodene er beregnet ut i fra avstanden fram til midtpunktet i elementet (projeksjonen av midtpunktet ut til buen), mens avstanden fra midten i elementet og til nodene er beregnet vha lengden på det rette elementet. Dvs. at for elementet øverst i figur A.121 vil de to nodenes plassering beregnes veldig feil, siden buelengden mellom nodene nesten er det dobbelte av elementlengden.

Heldigvis er de andre elementene mye bedre tilnærminger av buen, slik at her blir feilene små. Figur ?? viser at verdien “**s**” (som er avstanden tilbake til 1. *joint* langs buen) er bare 6 millimeter feil for siste node.

Siden “**s**” er beregnet feil har det skapt feil i to beregninger:

- Momentfaktoren  $C_{my}$ , som benyttes i ståldimensjonering av bjelkesøyler i hht. tabell B.3. i EC 3-1-1. [2]
- Lumping av *line loads* beregnes feil fordi man henter lastintensitet fra feil punkt, med feil momentarm.

**NB:** Dersom det brukes en normalt fin elementinndeling vil buelengdene og elementlengdene bli så godt som like. Derfor vil ikke brukeren merke feilene med mindre man bruker en grov elementinndeling. Som ingeniør bør man uansett være klar over at en fin elementinndeling er hensikts-

messig i buer. Dermed skjer det i praksis ikke store feil i beregningene, men det er viktig å være klar over at de tilnærmede metodene som brukes i **fap2D** har et visst gyldighetsområde.

### Endring

Verdien “**s**” beregnes nå riktig. Dvs. at presentasjon av resultater ikke lenger er forvirrende for brukeren.

Verdien brukes fortsatt i beregningen av  $C_{my}$  og *line loads*. For laster som virker normalt på elementer vil “**s**” være korrekt å bruke i beregningene. Om lastene derimot ikke står normalt på vil det skape en viss feil, fordi projeksjonen fra elementet og til buen ikke blir den korrekte verdien å hente ut lastintensitet fra. Se figur A.123 for en illustrasjon (NB: Det har ikke blitt tid til å studere *load lumping*-prosedyren i detalj, så vi kan ikke garantere at vi har tenkt helt riktig i vår påstand, men vi frykter at det er slik).

### Argumentasjon

Det var viktig å rette denne feilen, fordi brukeren må få konsistent informasjon. Dessuten er det kritisk at beregningsmetodene i programmet er korrekte (eller meget gode tilnærminger).

Feilen ble oppdaget sent i masteroppgaven og måtte rettes kjapt. Derfor ble det bestemt at man ikke skulle legge vekt på å gjøre om på beregningen av  $C_{my}$  og *line loads*. Bruken av *arch members* forutsetter uansett en fin elementinndeling og i praksis vil man da ikke oppleve merkbare avvik.

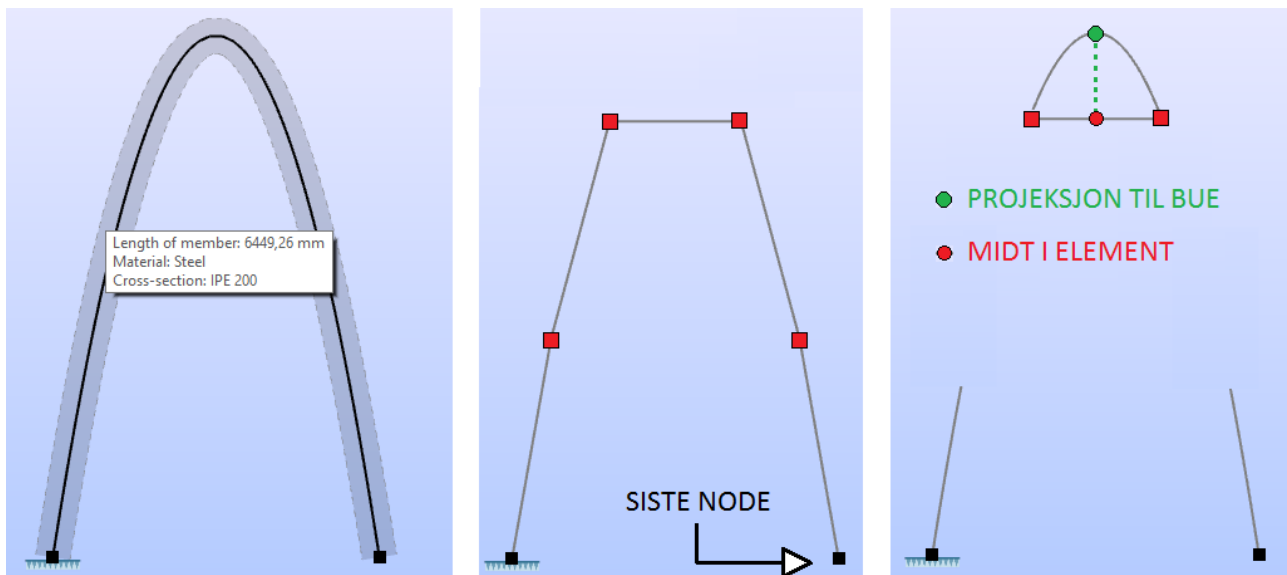
### Metadata

ID:	31
Klassifisering:	Feil
Sist modifisert:	27. mai
Modifisert av:	Espen, Torjus
Rapportert i versjon:	3.0
Rapportert dato:	Vår 2013
Rapportert av:	Espen

### Endrede klasser

fap2D.Computations	NodeLoadList
	HelpNode
	Element
	DesignCalculations
	ComputationalModel.ModelTools
fap2D.Gui	FrmExportDialog
	FrmUtilizationDialog
	FrmSectionForcesDialog
	FrmInfluenceLineResultsDialog

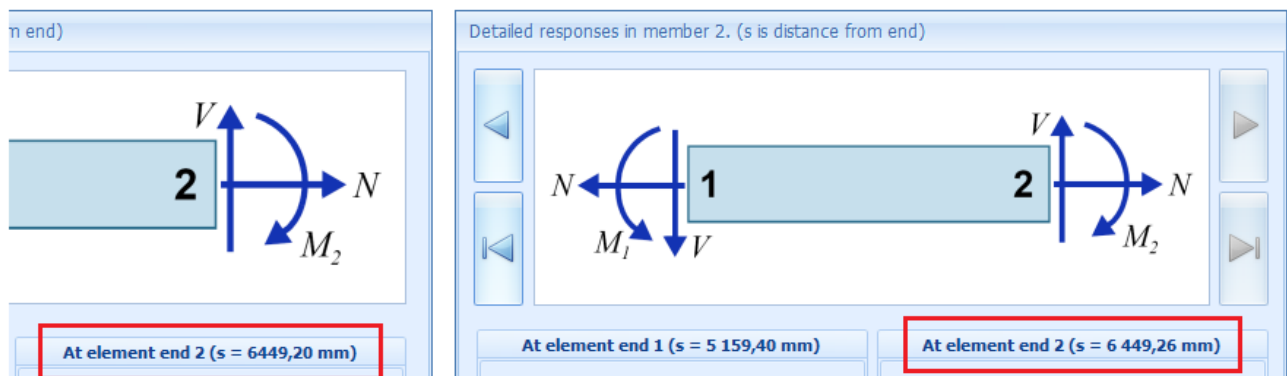
### Figurer



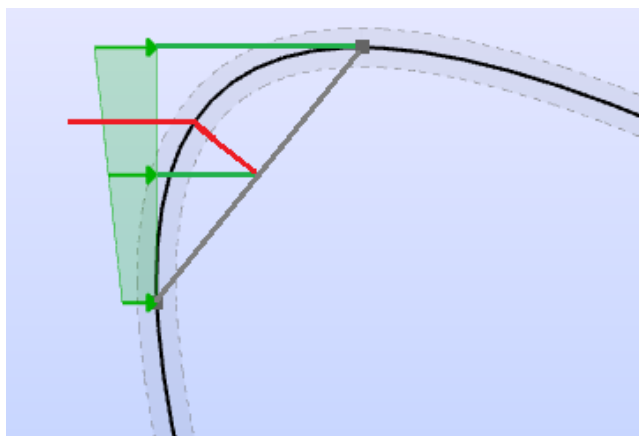
Figur A.121: *Arch member* som deles i fem elementer (bilde t.v. og i midten), der det øverste elementet blir vesentlig kortere pga. at buelengden mellom nodene skal vær den samme.

Avstanden fra 1. *joint* til siste node (= 2. *joint*) i *member*-en skal svare til lengden som vises når man holder over *member*-en (t.v), når resultater presenteres, for å unngå at brukeren må forholde seg til både buelengder og faktiske elementlengder.

Bildet t.h. viser hvordan midtpunktet i et element er tvetydig definert i **fap2D**. Den røde sirkelen er det faktiske midtpunktet (og SKAL brukes i beregninger), mens den grønne sirkelen er normalprojeksjonen av midtpunktet ut på buen. Sistnevnte brukes feilaktig i noen beregninger.



Figur A.122: **3.0** (t.v): Beregningen av “s” er 6 millimeter feil i forhold til lengden av buen til venstre i figur A.121. **3.1** (t.h): Beregningen er korrekt.



Figur A.123: Illustrasjon av mulig feilberegning av *line loads*.

### A.5.5 *Default load combination og time axis*

#### Endringer

For å utføre de ulike analysene i fap2D må man definere en aktiv *load combination* for analysen. For enkelte analyser er det angitt en *default load combination*, mens for andre analyser er det ingen *default*.

Noen av *default*-verdiene er vanskelige å være klar over. Blant annet vil egenvekt (*own weight*) bli lagt til i *default load combination* for lineær statisk analyse, dersom det ikke finnes andre laster i modellen.

For de analysene som ikke automatisk angir en *load combination* fikk brukeren før en feilmelding når en analyse blir forsøkt startet. Dette ødela programflyten og kunne forvirre brukeren fordi feilmeldingene ikke er presise for alle tilfeller.

Derfor ble det gjennomført noen endringer:

- *Linear static analysis*: *Default load combination* inneholder ikke lenger egenvekt, dersom det ikke er andre laster. Derimot blir man varslet via en dialog om at det ikke er satt på last, og man får mulighet til å inkludere egenvekt i analysen. Figur A.124 viser dialogen.

Det er i tillegg lagt til en innstilling i *settings* der brukeren kan velge om egenlasten alltid skal inkluderes dersom ingen laster er gitt.

- *Buckling analysis*: Dersom man ikke har angitt lastkombinasjon (vha. *loading dialog*) vises en dialog som gir brukeren mulighet til å åpne dialogen og aktivere en *load combination*. Denne endringen henger sammen med punkt A.5.1.
- *Nonlinear static*: Samme logikk som for *buckling analysis* er implementert.
- Dynamiske analyser: Alle dynamiske analyser (bortsett fra *earthquake response spectrum*), som har *loading* eller *time axis dialogs* vil vise en varselboks som guider brukeren til dialogene (når *dynamic load combination* eller *time axis* ikke er dinfert riktig).

#### Argumentasjon

Endringene gjør programmet mer konsekvent, på tvers av analyser. Dessuten blir man klar over hva som må gjøres og kan ordne problemene effektivt. Hensikten bak endringen er å minske

behovet for *tutorials* og leting i UM. Dessuten vil man fort læres til å gå innom dialogene, før man i det hele tatt trykker på *Run analysis*-knappen.

Innstillinen for egenvekt gjør programmet fleksibelt -brukeren får mer kontroll.

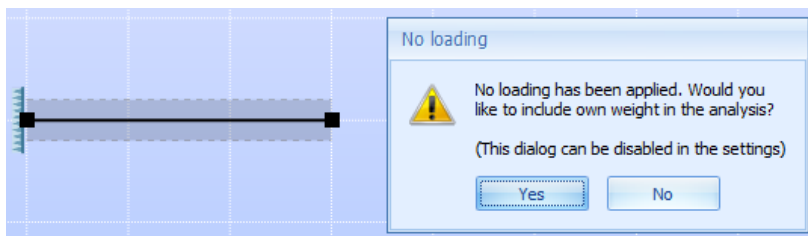
### Metadata

ID: 91  
 Klassifisering: Endring  
 Sist modifisert: 6. juni  
 Modifisert av: Torjus  
 Rapportert i versjon: 3.0  
 Rapportert dato: Vår 2013  
 Rapportert av: Kolbein

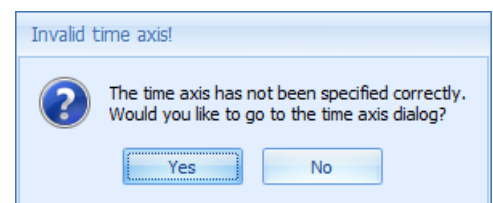
### Endrede klasser

fap2D.Gui FrmMainGui.AnalysisPage  
 fap2D.Computations ComputationalModel.ModelTools

### Figurer



Figur A.124: **3.1:** Dialogboks vises når det ikke er satt på laster i konstruksjonen. Analyse blir kjørt med egenvekt ("JA") eller ingen laster ("NEI").



Figur A.125: **3.1:** Dialogboks vises når *time axis* ikke er korrekt spesifisert for en dynamisk analyse.

## A.5.6 Konvertering av laststeg i *real time/influence line analyses* til statisk lasttilfelle

Disse to analysene har mulighet for å konvertere et laststeg i analysen til et statisk lasttilfelle som kan brukes videre i modellen til andre analyser.

### Feil

Dersom man prøver å legge til samme statiske lasttilfelle fra et steg i en *real time analysis* eller *influence line analysis* kan det oppstå feil plassering av lastene i lasttoget.

Grunnen er at prosedyren som skal opprette *internal joints* (som lastene plasseres i) kommer i konflikt med *joints* som allerede eksisterer i modellen.

Om man benytter lasttog (med tett plasserte laster) kan det derfor hende at lastene havner i samme *internal joint*, når man prøver å legge til det samme laststeget flere ganger.

Dersom man prøver å plassere en av lastene i lasttoget i et *joint* oppstår det konflikt fordi det blir forsøkt å legge til et *internal joint* der det er et vanlig *joint*, og det er ikke lov i **fap2D**. Figur A.129 illustrerer situasjonen.

### Endring

Lagt til en advarsel i *tooltips*-et til knappen *Convert to LC*, der det advares om å opprette mer enn ett statisk lasttilfelle i samme modell.

Sørget for at det ikke er mulig å angi flere lasttilfeller med samme navn. Det brukes nemlig standardverdi for navnet som brukeren ikke trenger å endre før man konverterer. Ved å sørge for at det ikke kan legges til flere med samme navn blir terskelen for å legge til flere statiske lasttilfeller høyere.

### Argumentasjon

Feilen skulle gjerne vært rettet ordentlig, men man har ikke fått tid til å gjøre det. Derfor er quasiløsningen over brukt for å minske faren for at feil oppstår.

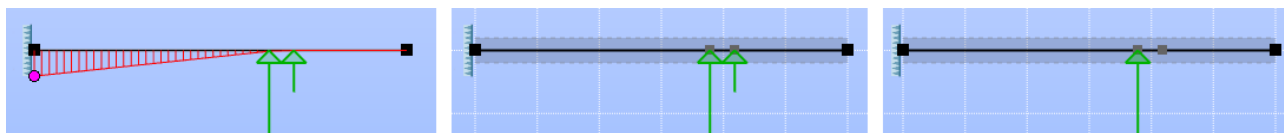
### Metadata

**Status:** Forbedret  
**ID:** 94  
**Klassifisering:** Feil  
**Sist modifisert:** 12. mai  
**Modifisert av:** Torjus  
**Rapportert i versjon:** 3.0  
**Rapportert dato:** -  
**Rapportert av:** -

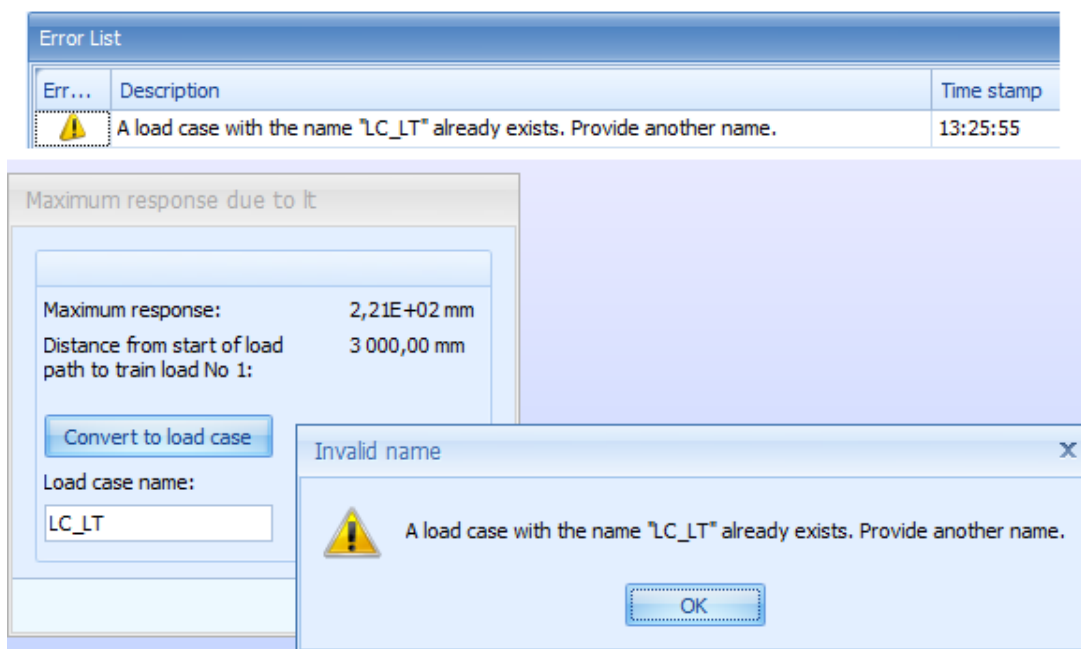
### Endrede klasser

fap2D.Gui FrmMainGui.ResultsPage  
 FrmXtrmDialog, .Designer, .resx  
 fap2D.Computations ComputationalModel.LoadTrain

### Figurer

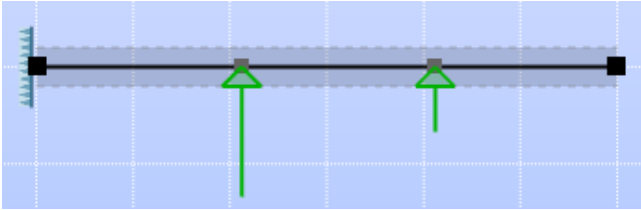


Figur A.126: **3.1:** Bildet t.v. viser et steg i en *influence line* eller *real time analysis*. Bildet i midten viser opprettelsen av et statisk lasttilfelle (basert på laststeget i bildet t.v.). Bildet t.h. viser hva som skjer når man prøver å legge til det samme lasttilfellet på nytt: Den ene lasten blir borte. Avstanden mellom lastene er  $200\text{mm}$ .

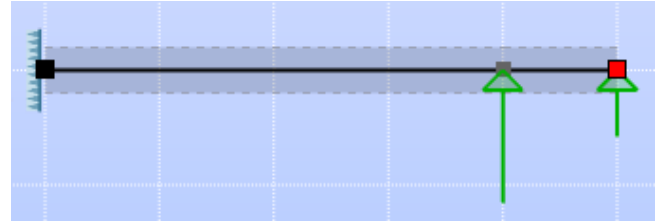


Figur A.127: **3.1:** Figuren illustrer at dersom avstanden mellom lastene i lasttoget er tilstrekkelig stor vil det ikke oppstå noen feil. Avstanden er ca  $1000\text{mm}$  mellom hvert *joint*.





Figur A.128: **3.1:** Dersom avstanden mellom *joints* er tilstrekkelig oppstår det ingen konflikter.



Figur A.129: **3.1:** Dersom man prøver å konvergere en last som står i et vanlig *joint* vil det forsøkes å opprette et *internal joint*.

## A.6 Resultater

### A.6.1 Fortegn for maksimale aksialkrefter

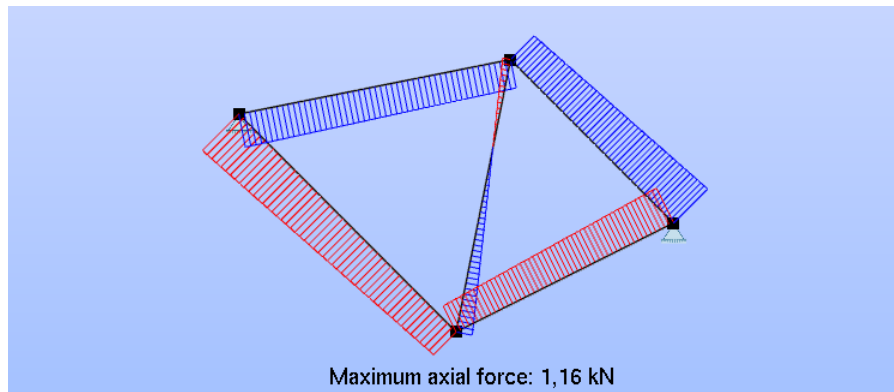
ID	24
Klassifisering	Nytt
Modifisert dato	26.02
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	-
Rapportert av	-

### Endringer

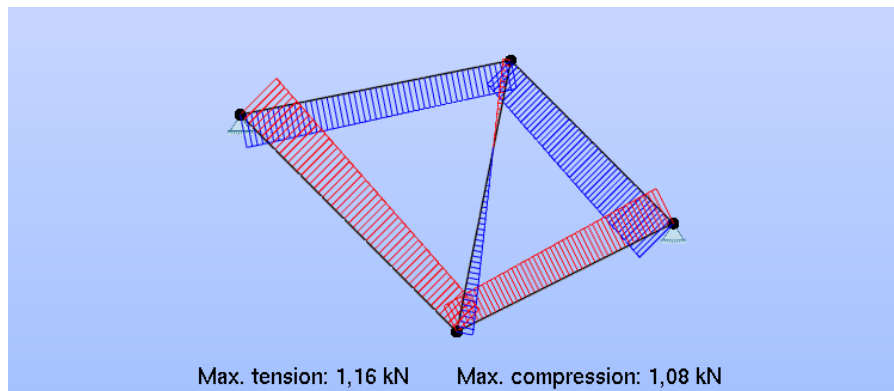
Aksialkrefter beskrives enten som strekk eller trykk, og i beregningene til **fap2D** har de hhv. positivt og negativt fortegn. Siden strekk og trykk har ulik betydning for en konstruksjon er resultatvinduet endret til å vise den største verdien for hver av dem, ikke bare den største absoluttverdien til aksialkreftene.

Teksten er plassert på én linje slik at den ikke dekker over diagrammet, og slik at ekstremalverdiene vises på en konsistent måte for alle resultater. Det er valgt å beskrive kreftene med navn istedenfor fortegn for å unngå forvirring.

<i>Namespace</i>	Endrede klasser
fap2D.Computations	ComputationalModel
	ComputationalModel.Drawing
	ComputationalModel.ModelTools



Figur A.130: Før endring - kun maksimal absoluttverdi.



Figur A.131: Etter endring - maksimalt trykk og strekk.

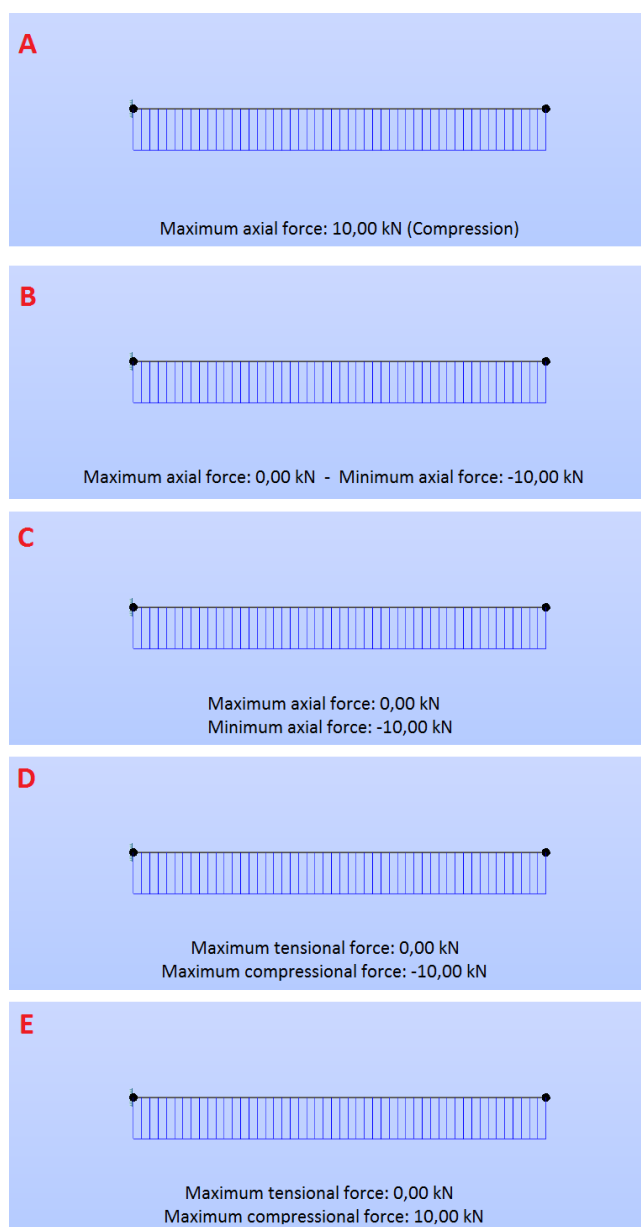
### Detaljer

Før var maksimal aksialkraft beregnet som den største absoluttverdien av aksialkreftene. Det er viktig å skille mellom trykk og strekk.

Maks aksialkraft kan være både strekk og trykk, og begge er interessante verdier. Følgende avgjørelser ble tatt:

- Både trykk og strekk skal vises
- Teksten skal være fordelt på én linje for å ikke ta plass fra diagrammet
- Beskrivelse i form av *compression* og *tension* skal brukes istedenfor fortegn

Avgjørelsen er tatt i samarbeid med veileder, og på bakgrunn av figur A.132.



Figur A.132: Forslag brukt i vurderingen om utseendet til maksimal aksialkraft.

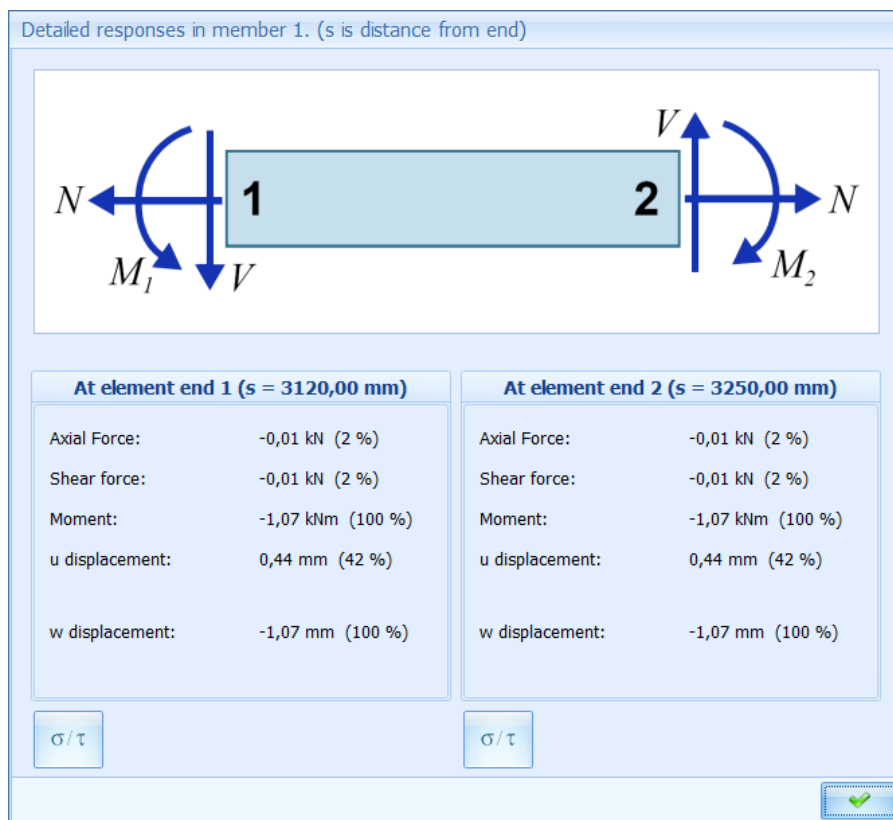
## A.6.2 Vektorlengde for forskyvninger

ID	65
Klassifisering	Nytt
Modifisert dato	05.03
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	04.03
Rapportert av	-

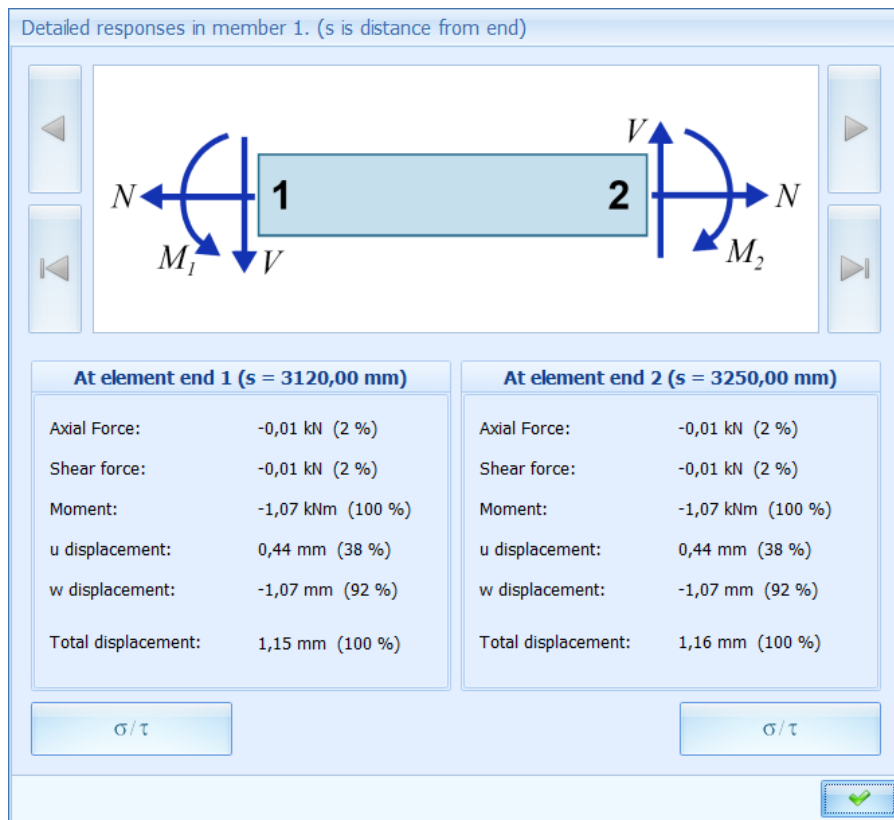
## Endringer

Maksimal forskyvning er endret fra å være den største forskyvningen langs en av aksene, dvs. maksimal u-forskyvning (langs x-aksen) eller maksimal w-forskyvning (langs z-aksen) til å være den største kombinerte vektorlengden av forskyvningene.

<i>Namespace</i>	Endrede klasser
fap2D.Computations	ComputationalModel ComputationalModel.Drawing ComputationalModel.ModelTools
fap2D.Gui	FrmSectionForcesDialog



Figur A.133: Før endring - kun u- og w-forskyvning vises - w er maksimal forskyvning.

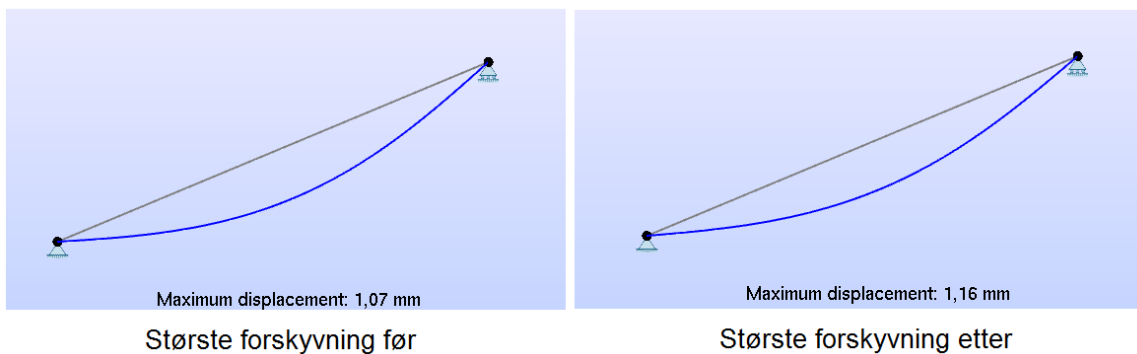


Figur A.134: Etter endring - total forskyvning er lagt til og er valgt som maksimalverdi.

## Detaljer

I dialogboksen for elementresponsen er det gitt en prosentverdi for forskyvning av den valgte elementnoden i forhold til maksimalverdien. Nå benyttes den maksimale kombinerte vektorlengden som 100%, og hver forskyvningskomponent sammenlignes med denne.

Endringene er synlige i diagrammet for forskyvninger, og i dialogboksen for elementresponsen.



Figur A.135: Eksempel - Den største komponenten av deformasjonene (t.v.) er mindre enn den største vektorlengden av deformasjonene (t.h.)

### A.6.3 Navigering av elementresultater

ID	66
Klassifisering	Nytt
Modifisert dato	06.03
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	04.03
Rapportert av	-

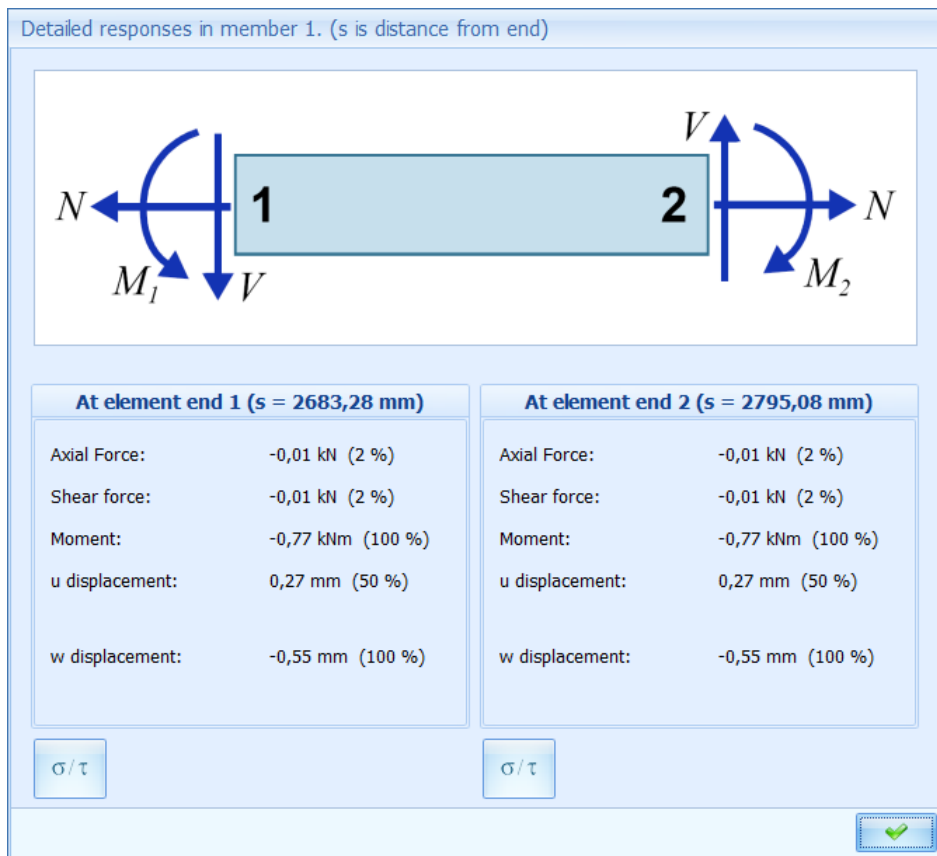
#### Endringer

Det er implementert navigering mellom elementene i dialogboksen for resultatene slik at det er enklere å finne elementet på en bestemt posisjon.

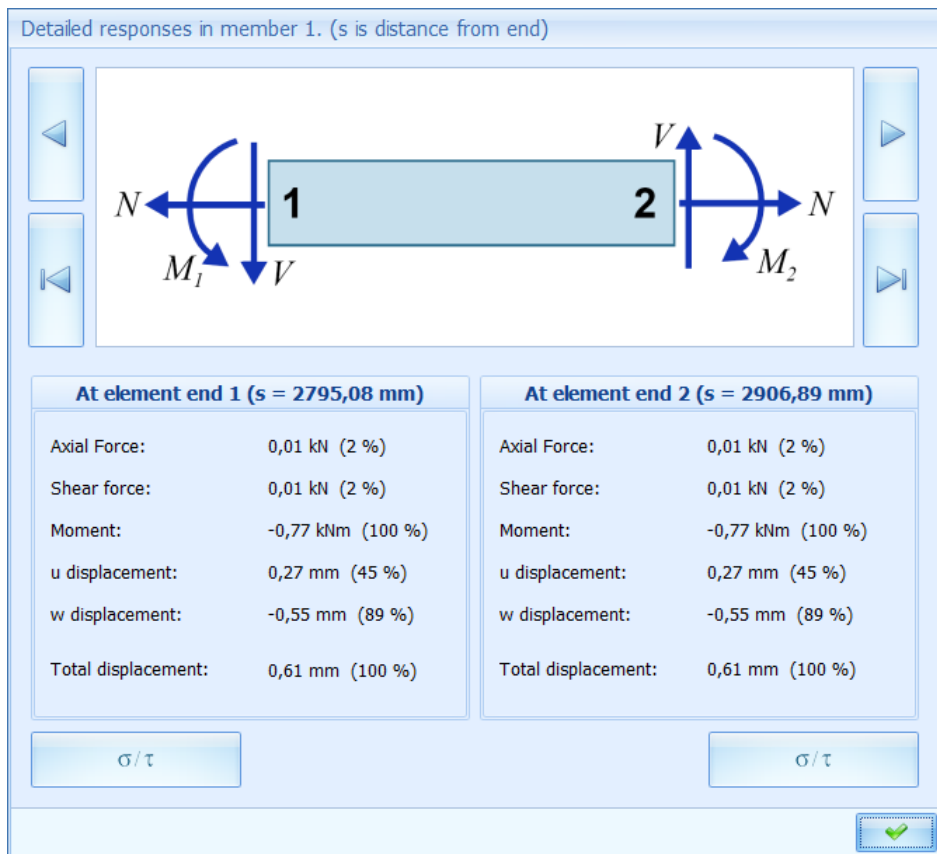
Det er valgt å legge til knapper i dialogboksen for resultatene for å gå til forrige, neste, første og siste element langs den aktuelle *memberen*. Som vist i figur A.138 er det inkludert tastekombinasjoner som tilsvarer navigasjonsknappene for å gjøre brukeropplevelsen bedre.

For å gjøre navigeringen mest mulig intuitiv skjer navigering kun internt i en *member*, på tvers av *submembers*. Det betyr at første og siste element håndteres spesielt. Det er implementert *properties*, *NextElement* og *PreviousElement*, for å referere til naboelementer. Naboelementet til et element som ligger på enden av en *member* er valgt til å være seg selv (*this*) for å unngå *nullpointers*.

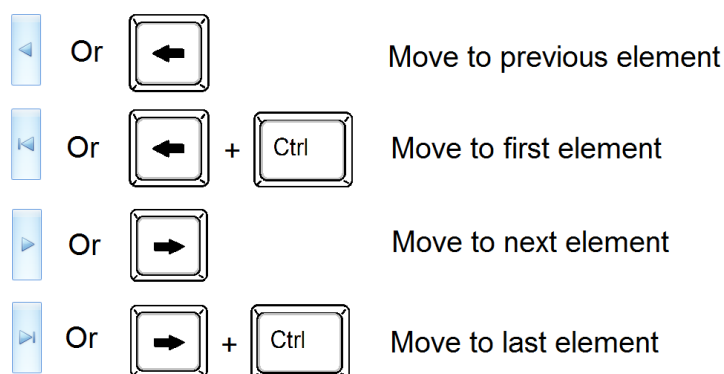
<i>Namespace</i>	Endrede klasser
fap2D.Computations	ComputationalModel.ModelTools Element
fap2D.Gui	FrmSectionForcesDialog



Figur A.136: Før endring



Figur A.137: Etter endring - Nye knapper for navigering der de tar opp minst plass.



Figur A.138: Navigeringskontroller - knapper og tilsvarende tastaturkombinasjoner

## Detaljer

### Endringer i brukergrensesnitt

Mindre endringer er nødvendig for å implementere denne funksjonaliteten:

- Knapper for navigering til forrige/neste element må legges til
- Piltastene på tastaturet må tilsvare navigeringsknappene
- Knapper for navigering til første og siste element på *memberen* må legges til

Dersom navigering til første og siste element blir implementert bør det være taster for disse. Det foreslås *Ctrl* + venstre/høyre piltast.

**Merk:** Også konstruksjonen i modellvinduet må oppdateres for å vise hvilket element som er markert til enhver tid.

### Endringer i kode

Elementnummereringen gjør det utfordrende å finne nærliggende elementer, derfor må struktur for slik navigering opprettes ved *meshing*. Elementene er implementert slik at de er frittstående, uten en øvre struktur. De har imidlertid en variabel for *MemberID* for *memberen* som opprettet elementet.

For å gjøre navigeringen mest mulig intuitiv skal navigering kun skje internt i en *member*, på tvers av *submembers*. Det betyr at første og siste element må håndteres spesielt. Naboelementet til et element som ligger på enden av en *member* kan være:

- null
- seg selv (*this*)
- elementet på andre enden av *memberen*

Det mest intuitive er å sette det til null eller seg selv og implementere knapper for å gå direkte til elementet i hver ende av *memberen*. Det er valgt referere til seg selv kan hjelpe til å unngå feil i koden.

Hvilket element som regnes å ligge før et annet element er bestemt av retningen til den bakenforliggende *memberen*. Elementet ved *jointet* i ende 1 vil være det første elementet, og elementet ved *jointet* i ende 2 vil være det siste elementet. For at første element skal ligge til venstre



for siste element blir den bakenforliggende *memberen* tilpasset den mest intuitive retningen. Se endring A.4.14 *Member*-retning for detaljer.

#### A.6.4 Forbedret visning av spenninger i elementer

ID	67
Klassifisering	Nytt
Modifisert dato	12.03
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	Høst 2013
Rapportert av	Espen

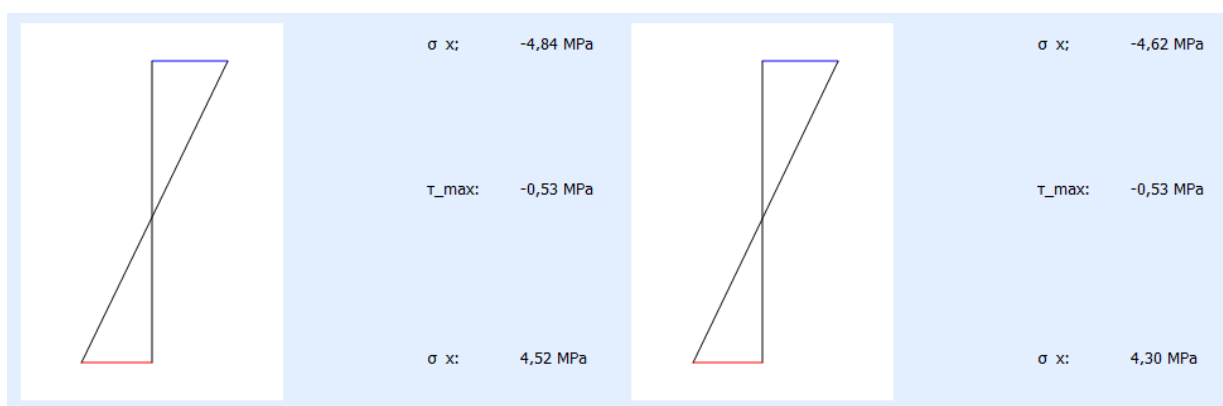
#### Endringer

Implementert forbedret visning av spenninger i resultatvinduet. Ny visning reflekterer hvilken node av elementet man ser på, og det er mer intuitivt hva som er trykk og hva som er strekk.

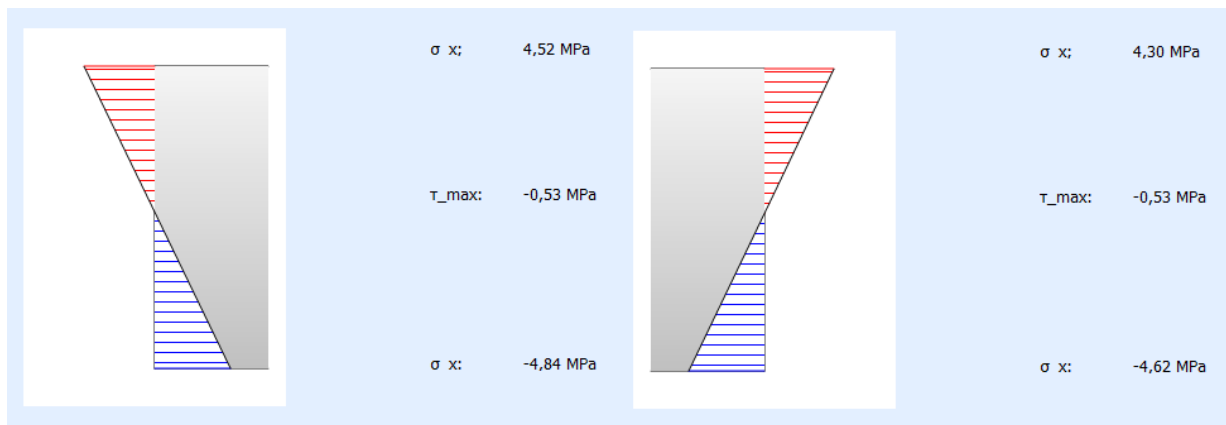
Det nye diagrammet bruker mye av den gamle koden, men viser tre endringer:

1. et utsnitt av enden av elementet
2. skravert diagram
3. intuitiv retning av diagrammet

<i>Namespace</i>	Endrede klasser
fap2D.Gui	UC_Stresses FrmStressDialog



Figur A.139: Før endring - spenningsdiagrammer for hhv. venstre og høyre ende av et element



Figur A.140: Etter endring - spenningsdiagrammer for hhv. venstre og høyre ende av et element

## Detaljer

Det var ønske om å forbedre visning av spenninger i resultatvinduet. Ny visning reflekterer hvilken node av elementet man ser på, og viser tydelig hva som er trykk og hva som er strekk.

UC\_Stresses.cs var litt uryddig skrevet, og krevde derfor en del refaktorisering underveis. Fortegn og indeksering av krefter/momenter førte til forvirring og nye feil, derfor tok implementeringen lenger tid enn forventet.

Å vise hvilken ende av elementet man ser på hjelper for å skille mellom strekk og trykk. Skravert diagram er i takt med måten man tegner diagrammer i litteraturen.

## A.6.5 Visning av symbol for maksimalverdier i resultatdiagrammer

### Nyhet

Lagt til et sirkelsymbol i resultatdiagrammene som viser hvor den maksimale responsen i diagrammet oppstår. Dette gjelder:

- Displacements
- Moment
- Axial force
- Shear force
- Steel design usage

Det var en relativt enkelt prosess å gjennomføre endringen, men man glemte i første omgang å ta hensyn til at elementer kan ha eksentrisitet og derfor ble symbolet forskjøvet litt. Dette ble enkelt ordnet senere. Dessuten ble skapt en *exception* når det ikke var noen maksverdi (Max = 0) i diagrammene, men denne ble også rettet.

Det er lagt til en knapp i *toolbox* som kan brukes til å slå visningen av makssymbolet av/på. Standardinnstillingen er "på".

### Endrede klasser

fap2D.Gui	FrmModel
	FrmMainGui.Toolbox
fap2D.Computations	ComputationalModel.Drawing
	ComputationalModel
	ComputationalModel.ModelTools
fap2D.Utilites	GlobalSettings.ComputationalModel
	OpenGLUtils
fap2D.StructuralData	Joint
	Member

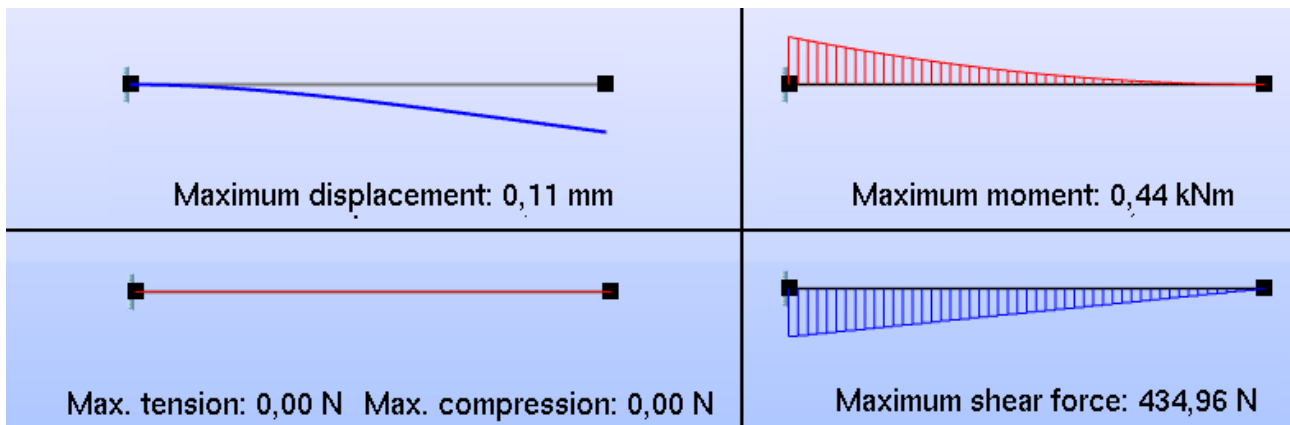
### Figurer

### Argumentasjon

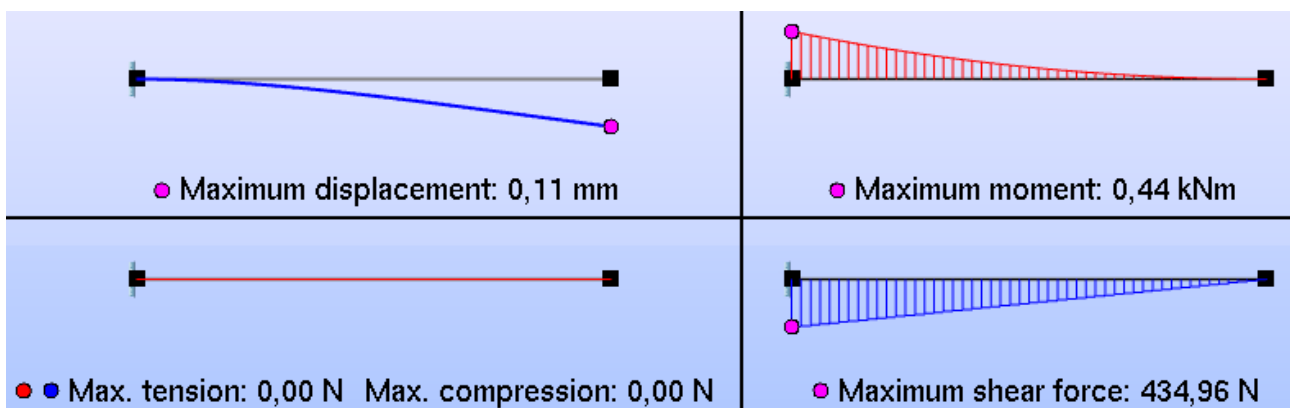
Det er veldig nyttig å se hvor maksimalresponsen i diagrammene oppstår, uten man er nødt til å lete seg frem. Viktig informasjon å vise i en analyserapport.

### Metadata

ID:	58
Klassifisering:	Nytt
Sist modificert:	31. mai
Modifisert av:	Torjus, Espen
Rapportert i versjon:	3.0
Rapportert dato:	Vår 2014
Rapportert av:	Kolbein, Torjus, Espen



Figur A.141: **3.0**: Resultatdiagrammene viser ikke plasseringen av maksimalverdiene.



Figur A.142: **3.1**: Resultatdiagrammene viser plasseringen av maksimalverdiene. Mulig å slå visningen av/på.

## A.7 Steel design

### A.7.1 Feilmelding ved *Component control*

ID	3	Rapportert i versjon	3.3.0 - DE
Klassifisering	Feil	Rapportert dato	Vår 2013
Modifisert dato	21.05	Rapportert av	Espen
Modifisert dato	Torjus	Endrede klasser	
<i>Namespace</i>		FrmMainGui.AnalysisPage	
fap2D.Gui		fap2D.Computations	
		ComputationalModel.Analyses	

#### Feil

Dersom man forsøker å utføre component control for en konstruksjon der det er veldig lave eller ingen aksialkrefter vil kontrollen avbrytes med følgende feilmelding: Frame2D\_LB: Buckling analysis aborted due to very low level of compression forces."

**Endring**

Grunnen til feilmeldingen er at component control utfører dimensjonering i hht. bøyingsknekking ingen EC 3-1-1 punkt 6.3. Det kan være vanskelig å se sammenhengen, så det bør legges til en mer forklarende feilmelding. Feilen er omtalt i [10, s. 14].

Det viste seg at man må gjøre endringer i Frame2D for å få endret den aktuelle feilmeldingen som skrives ut. Derfor løste man dette problemet ved å lage en ekstra feilmelding som skrives ut samtidig som den opprinnelige. Feilmeldingen ser ut som følger:

*“Component steel design control aborted because linearized buckling is not relevant. Elastic steel design is sufficient.”*

**A.7.2 Dårlig fargeskala for steel design usage****Metadata**

ID:	16
Klassifisering:	Feil, Endring
Sist modificert:	20. mai
Modifisert av:	Torjus
Rapportert i versjon:	3.0
Rapportert dato:	Høst 2013
Rapportert av:	Kolbein

**Endrede klasser**

fap2D.Computations	Element
fap2D.Gui	FrmMainGui.ResultsPage
	FrmSettings
	FrmModel.OpenGL
fap2D.Storing	StoringUtilities
fap2D.Utilities	GlobalSettings
	global.fap

**Feil**

Fargeskalaen som brukes for å visualisere steel design usage i hht. dimensjonering i EC 3-1-1 er ikke heldig. Fargeverdiene som vises interpoleres mellom fargeverdien i hver av nodene i et element. Dette er uheldig fordi det kan gi inntrykk av feil utnyttelse.

Fargen rød brukes for å vise når utnyttelsen er over 100 % av kapasiteten. Rødfargen interpoleres ned til en gulfarge (for verdier mellom 67 - 100 %). Dette gjør at utnyttelsen på 90 % får en tilnærmet rød farge. Det er altså ikke mulig å se at denne utnyttelsen faktisk er OK.

**Endring**

Tidligere var det brukt en fargeskala der man kunne angi sine egne fargeskalaen. Denne er nå erstattet med en statisk fargeskala som består av 11 farger. Rød er for alt over 100 %, mens de andre er for hver 10 % mellom 0 og 100. Fargeverdiene interpoleres ikke lenger mellom noder, den er konstant innad i hvert element og det er en mye tydeligere og korrekt visualiseringsmåte. Fargen innad i elementet baseres på lineær interpolering mellom utnyttelsen i nodene i elementet, med mindre en av nodene har mer enn 100 % utnyttelsen - da er fargen alltid rød!

Algoritmen for å berene fargeverdiene i elementene er dermed endret/rettet. Skalaen som vises i resultatvisningen er gjort bedre. Dessuten er visualiseringen tegnet bredere (over hele tverrsnittshøyden).

**Argumentasjon**

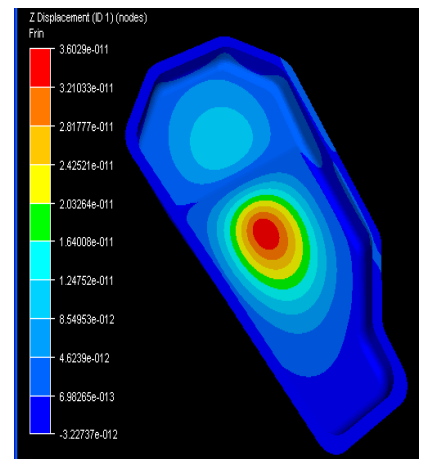
Feilen er ganske grov i og med at resultatene kan feiltolkes helt. Riktignok kan man navigere gjennom alle elementene og sjekke utnyttelsen, men det er litt mer tungvint. Dessuten er fargevisualiseringen det første som møter en etter kontroll så den må være korrekt.

**Algoritme for color mapping**

Hentet fra forelesningsnotater i TPG4162 3D-visualisering av petroleumsdata, v. Stein Dale.

# Scalar Algorithms: Color Mapping

- Color Mapping
  - values represented by color scheme
  - color scheme based on interpolation
- Intuitive colors are important
  - e.g. temperature from blue (min) to red (max)
  - Usually non-overlapping sections
- Color scheme should be user selectable
  - Determining places in Norway with more/less than 2 m rain/year by red/green is easier than using shades of grey
  - Legend should sometimes be non-linear (logarithmic, etc)



### A.7.3 Tegning av elementresultater

#### Metadata

ID: 73  
Klassifisering: Endring  
Sist modifisert: 20. mai  
Modifisert av: Torjus  
Rapportert i versjon: 3.0  
Rapportert dato: Høst 2013  
Rapportert av:

#### Endrede klasser

fap2D.Gui FrmModel.OpenGL  
fap2D.Computations Element

#### Endring

Før ble utnyttelsen i hht. steel design tegning veldig smalt slik at det er vanskelig å se hva utnyttelsen.

Derfor tegnes nå utnyttelsen over en bredde som er lik tverrsnittshøyden på elementet. Dette gir også en bedre assosiasjon til konstruksjonens form og kan si noe om hvorfor fordelingen av utnyttelsen er som den er. Mer lettlest og intuitiv visualisering nå.

### A.7.4 Bruk av steel design control i ulike analyser, og for ulike membertyper og cross sections.

#### Metadata

ID: 93  
Klassifisering: Feil, Endring  
Sist modifisert: 27. mai  
Modifisert av: Torjus  
Rapportert i versjon: 3.0  
Rapportert dato: Vår 2014  
Rapportert av: Torjus

#### Endrede klasser

fap2D.Gui FrmMainGui.ResultsPage  
fap2D.Computations ComputationalModel.Analyses

#### Endring

Steel design controls har vært mulige å gjennomføre for både statiske og dynamiske analyser. Det har blitt bestemt, i samråd med veileder, at steel design kun skal være tilgjengelig for lineær statisk og ikkelineær statisk analyse heretter. Grunnen er at kontrollene ikke er beregnet å brukes for dynamiske beregninger, de er ikke gyldige.



Steel design controls er bare gyldige for visse members og cross section, i hht. UM [7, s. 70]. Det er nå lagt til dialogbokser som informerer brukeren om dette og avbryter kontroller når konstruksjonene ikke tilfredstiller kravene.

### Argumentasjon

Steel design er generelt lite testet og bør ikke brukes i stor grad, i alle fall ikke i feil type analyser, member eller cross section.

## A.7.5 Reaktivering av *Component control*

### Metadata

ID:	99
Klassifisering:	Feil
Sist modifisert:	20. mai
Modifisert av:	Torjus
Rapportert i versjon:	3.0
Rapportert dato:	Vår 2014
Rapportert av:	Kolbein

### Endrede klasser

fap2D.Computations	Element
	ComputationalModel.Analyses
	ComputationalModel.Drawing

### Feil

Det har vært brukt elastic steel design i stedet for component steel design control ved en feiltagelse, når man trykker på component-knappen. Reaktivert igjen nå.

## A.8 Eksport og import

### A.8.1 Eksport

ID	119
Klassifisering	Nytt
Modifisert dato	14.05
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	
Rapportert av	K. Bell

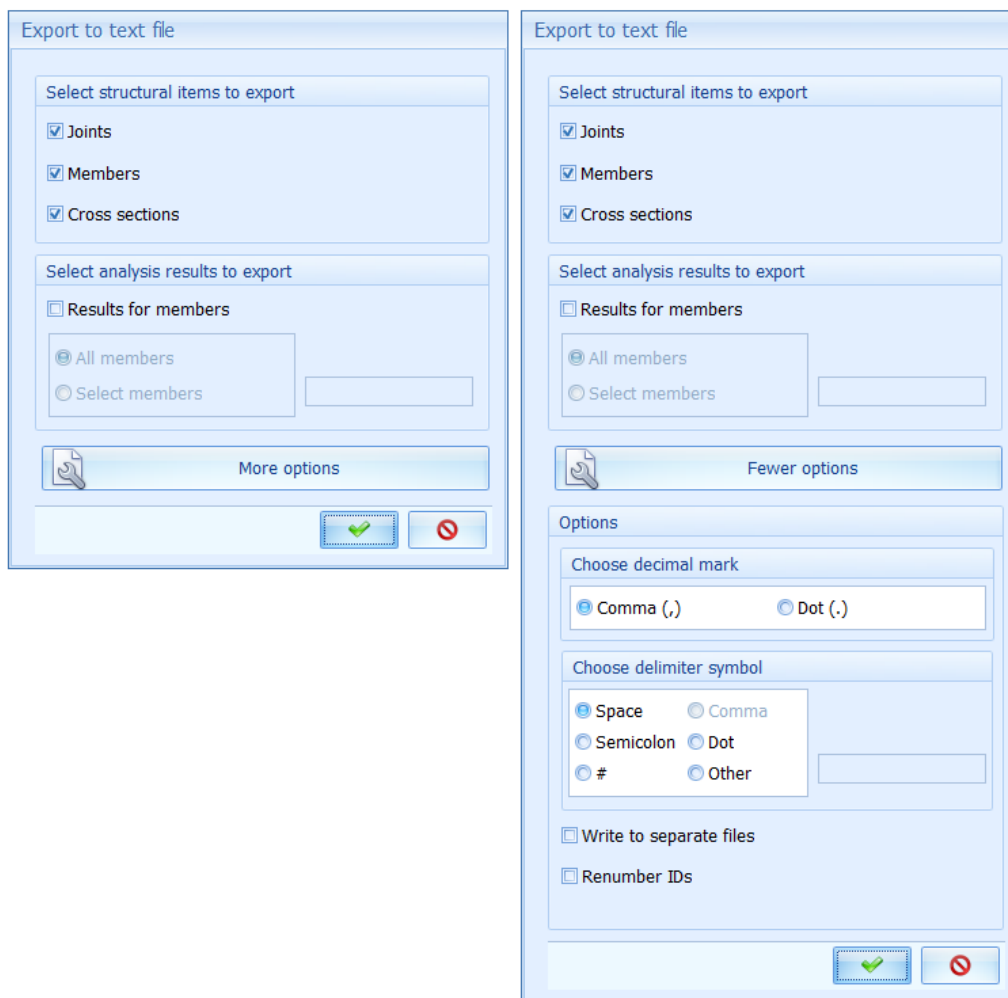
## Nytt

Det er lagt til funksjonalitet for å eksportere en åpen konstruksjonsmodell til et tekstdokument. Funksjonaliteten åpner for å eksportere deler av modellen til andre programmer, og kan fungere som en enkel rapportgenerator.

Følgende kan eksporteres:

- *Joint*-koordinater
- *Member*-informasjon
- Tverrsnittsdata
- Tverrsnittskrefter for valgte *members*

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmMainGui.ApplicationMenu.cs FrmExportDialog.cs (ny)



Figur A.143: Dialogboks for eksport – hhv. vanlig og utvidet.

fap2D version 3.5.2 - DE

Joint coordinates

Joint	X-coord. (mm)	Z-coord. (mm)
1	; 0	; 0
2	; 3000	; 0

Member info.

Member	Type	Geometry	Radius/height (mm)	Joints	Material	Cross sect.
1	; StraightBeam	; Straight	; 0	; 1, 2	; Steel	; IPE200, Rec 100x100

Cross section info.

Name	Type	A (mm <sup>2</sup> )	I (mm <sup>4</sup> )	ht (mm)	hb (mm)
IPE 200	; Predefined	; 2850	; 1,94E+07	; 100	; 100
Rec 100x100	; Parametric	; 10000	; 8,3333E+06	; 50	; 50

Results for members (section forces).

Results for member 1

Member type: StraightBeam  
 Load combination: Default load combination  
 End joints: 1, 2

s (mm) =	N (kN)	V (kN)	M (kNm)
-5,5565E-15	; 0	; -4,2895	; 5,7925
300	; 0	; -3,642	; 4,5057
600	; 0	; -3,0408	; 3,4131
900	; 0	; -2,4858	; 2,5008
1200	; 0	; -1,9771	; 1,7551
1500	; 0	; -1,5146	; 1,162
1800	; 0	; -1,0984	; 0,70759
2100	; 0	; -0,7284	; 0,37807
2400	; 0	; -0,40467	; 0,15955
2700	; 0	; -0,12718	; 0,038154
3000	; 0	; -0,12718	; 0

Figur A.144: Eksempel på eksportert tekstfil.

## Detaljer

### Brukeropplevelse:

1. Åpner dialogen for å eksportere informasjon.
2. Brukeren velger hvilke alternativer som skal eksporteres.
3. Brukeren velger desimaltegn og skilletegn.
4. Bruker velger om filene skal skrives til én fil eller flere separate filer.
5. Brukeren velger om *joints* og *members* skal renummereres før eksport.

### Eksportalternativer:

Eksportalternativene er valgt i samråd med veileder.

*Joint*-koordinater: Lister opp alle alle *joints*. Informasjonen som vises er: ID, x-koordinat og z-koordinat.

*Member*-informasjon: Lister opp alle *members*. Informasjonen som vises er: ID, *member*-type, geometri, høyde/radius, ID-en til de to *joints* som *member*-en er knyttet til, materiale og tverrsnitt.

Tverrsnittsdata: Lister opp alle tverrsnitt. Informasjonen som vises er: Navn, type, areal, 2. arealmoment, og avstand fra nøytralakse til øverste og nederste fiber i tverrsnittet.

Tverrsnittskrefter for valgte *members*: Resultater for hver elementnode i én eller flere *members*. Informasjon som vises er: *member*-type, lastkombinasjon, *joint* 1 og 2, plasseringen av snittet langs en av *memberens submembere*, og krefter i snittet (aksialkraft, skjærkraft og bøyemoment).

**NB:** Siste node i hvert element vil vise samme aksial- og skjærkraft som nest siste node fordi de henter krefter fra det samme elementet. Aksial- og skjærkraft er nemlig konstant i elementer. Derimot er momentet ulikt, siden det varierer lineært i elementer.

### Formatering

Utskriften skjer til én eller flere tekstdokumenter (.txt), i ASCII-format.

Faste kolonnebredder er valgt for at utskriften skal være lettleselig. Kolonnebredden tar hensyn til antall signifikante siffer i hver kolonne. Det ble valgt å bruke fem signifikante sifre for tallverdiene som eksporteres, noe som burde være presist nok for de fleste parametere.

Formatet på utskriften er koordinert med import-funksjonaliteten i A.8.2 **??**. Hvert dokument har en *header*-linje som **fap2D 3.1** og senere versjoner kan bruke for å tilpasse eksport og import etter versjonsnummeret, dersom eksport/import endres i fremtiden.

Man kan velge ulike skilletegn, enten forhåndsdefinerte eller egendefinert. Det er derimot ikke mulig å velge samme tegn for desimaltegn og skilletegn, for *parser*-en som tolker input-filene for import er ikke avansert nok til å takle dette.

For alle tallverdier skrives det ut enheter slik at det ikke er noen som helst forvirring om størrelsen på dem. For tverrsnittsdata skrives alle enheter ut med millimeter som grunnenhet, mens for resten følges enhetene som er angitt i *settings*.

Figur A.144 viser et eksempel på en eksportert fil.

## A.8.2 Import

ID	120
Klassifisering	Nytt
Modifisert dato	16.05
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	
Rapportert av	K. Bell

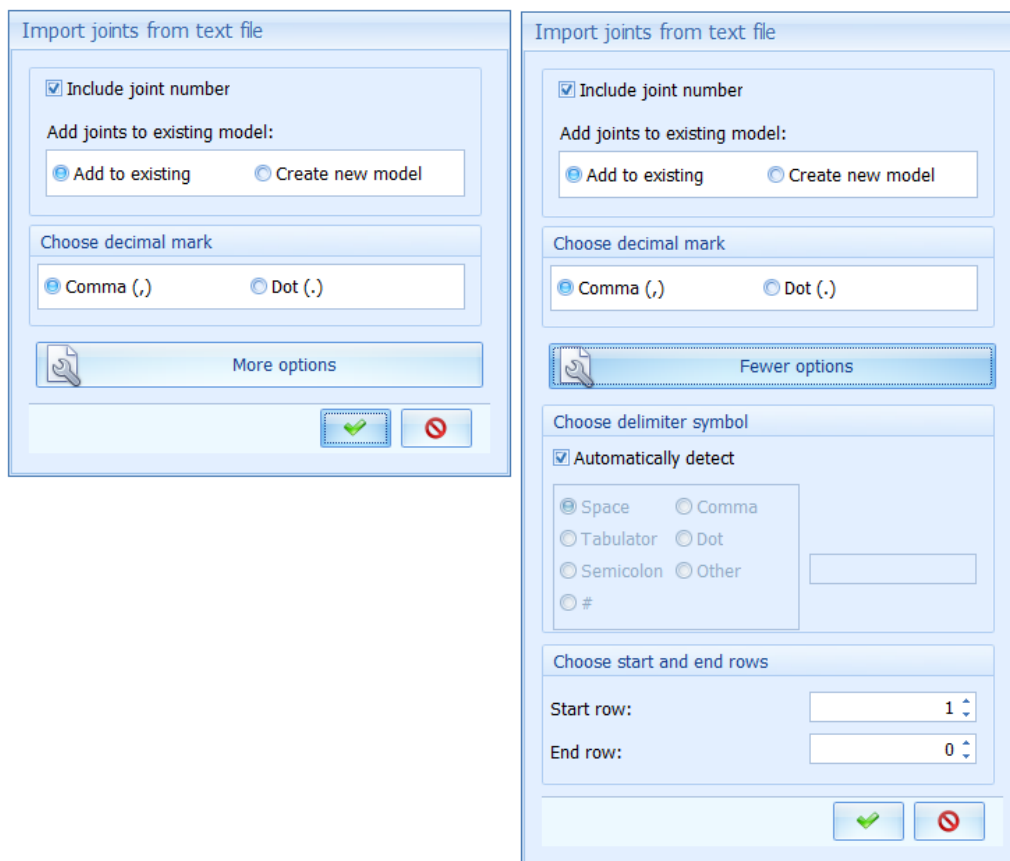
### Nytt

Det er implementert importfunksjonalitet for *joint*-koordinater. Import skal skje via et universelt format, derfor benytter denne funksjonaliteten tekstdokumenter. Siden **fap2D** også har en ny eksport-funksjon er import-funksjonen tilpasset formatet til en tilsvarende eksportert fil.

Brukeren velger filen som skal importeres og om *joints* skal legges til i en ny eller eksisterende modell. For å kunne laste inn *joint data* må man definere hvilket skilletegn som brukes for

dataverdiene i tekstfilen. Alternativt kan man la programmet prøve seg frem for å finne ut hvilket tegn som brukes.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmMainGui.cs FrmMainGui.ApplicationMenu.cs FrmImportDialog.cs (ny)



Figur A.145: Dialog med innstillinger for import – hhv. vanlig og utvidet.

## Detaljer

Å kunne importere *joint*-koordinater er nyttig for å kunne jobbe med samme modell i ulike programmer, samt sørge for enkel bakover-kompatibilitet til tidligere programversjoner. Ved en senere anledning kan funksjonaliteten utvides til å importere mer komplekse modeller.

### Brukeropplevelse:

Brukeren blir bedt om å velge fil som skal lastes inn i en fil-velger.

En dialog vil deretter be brukeren om å spesifisere hvordan dokumentet er bygget opp:

- desimaltegn (komma eller punktum)
- inkluder/ekskluder id-kolonne (hvis det eksisterer)
- legg til *joints* i eksisterende modell eller i en ny modell
- skilletegn (programmet prøver automatisk)

- startrad og sluttrad for dataverdier i tekstfilen

Det er mulig å skjule noen av innstillingene i dialogen. Brukeren har nemlig ikke behov for å justere alle innstillingene, i alle fall ikke hver gang den brukes.

En *smartparser* er laget som prøver flere ulike delesymboler på hver av radene. Dette valget kalles "*Automatically detect*", og søker gjennom hver symbol i hver aktuelle rad for å finne deletegn. Denne funksjonaliteten er noe krevende, dersom bruker har mange rader, derfor er det mulighet for å manuelt sette hvilket delesymbol som brukes.

**Merk:** Det forhåndsvalgte formatet skal være lik eksportformatet:

<Header>

```
Joint coordinates
Joint X-coord. Z-coord
  ii      xxxx   xxxx
  ..      ...    ...
```

**Skilletegn** Det gis ulike valgmuligheter som standard, i tillegg til å definere eget skilletegn. Mulighet for egendefinert tegn er begrenset til fem(5) tegn for å unngå misbruk. Tegnet kan ikke være det samme som desimaltegn. Bindestrek/minus tegn og plus tegn er heller ikke tillatt fordi de forbindes med fortegn.

Merk at komma (,) og punktum (.) håndteres avhengig av hvilket desimaltegn som brukes. Bruken av ", " og ". " gjør at man kan ha komma og punktum som både skilletegn og desimaltegn så lenge skilletegnet er omringet med mellomrom. Tilsvarende gjelder dette for fortegn. Legg merke til at mellomrom () og tabulator (" ") er lagt sist i listen over delesymboler, dette er fordi metoden `Split(...)` vil prøve alle symbolene i listen som deletegn, men prioriterer de første tegnene.

### Tallformat

Tall som importeres *parses* med `double.Parse(...)`, med unntak av `Id` som *parses* av `int.Parse(...)`. Merk at `double.Parse(...)` kan håndtere tall på eksponentform, f.eks. -1,2345E+11, et format som kan bli brukt av eksport om nødvendig.

## A.9 Settings

### A.9.1 Skrive *Global Settings* til *Appdata*

ID	122
Klassifisering	Endring
Modifisert dato	22.05
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	09.05
Rapportert av	

## Endringer

Endret filstien for *Global Settings*<sup>5</sup> til et sted bruker alltid har skriverettigheter til.

Endret også filstien for midlertidige backup filer som brukes av *undo/redo* funksjonaliteten.

<i>Namespace</i>	Endrede klasser
fap2D.Utilities	GlobalSettings.cs
fap2D.Storing	UndoRedo.cs

## Detaljer

### Global Settings:

Det er blitt advart mot å la programmet skrive til installasjons-mappen, og det er derfor besluttet å plassere det i en mappe som er avhengig av brukermiljøet på maskinen. Vanligvis skrives konfigurasjonsfiler til mappen:

```
C:\Users\\Appdata\Roaming\
```

og kan nås via den innebygde metoden:

```
Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData);
```

Denne mappen følger med bruker på tvers av maskiner dersom brukerprofilen eksisterer på et nettverk. På **NTNU** har studenter og ansatte en slik profil.

Det blir opprettet en under-mappe av **fap2D** i *Appdata*-mappen, slik at den endelige filstien til *Global Settings* blir:

```
C:\Users\\Appdata\Roaming\NTNU\fap2D\global.fap
```

### Log-file:

Analyserutinene i **Frame2D**-biblioteket skriver til en *log*-fil der kommentarer og data om analysen noteres. Denne filen skrives direkte til mappen **fap2D** ble startet fra. Dette inkluderer installasjons-mappen for vanlig bruk. Siden analyserutinene er skrevet i FORTRAN har ikke koden tilgang til miljøvariablene som peker til *Appdata*-mappen.

Backup-filer: Tidligere ble midlertidige filer skrevet til:

```
C:\Users\\Appdata\Roaming\fap2D\Temp\
```

For å samkjøre det med resten av *Appdata*-filene er dette nå endret til:

```
C:\Users\\Appdata\Roaming\NTNU\fap2D\Temp\
```

---

<sup>5</sup>*Global Settings* er en samling innstillinger for **fap2D**, ofte kjent som en konfigurasjonsfil.

## A.9.2 Redigert og fjernet *settings*

ID	76
Klassifisering	Endring
Modifisert dato	27.05
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	Vår 2013
Rapportert av	Kristian Pedersen, Torjus

### Endringer

Innstillingene som brukes i programmet er blitt revidert:

- Fjernet ubrukte og unødvendige *settings*.
- Laget *tooltips* som gir bedre forklaring av *settings*.
- Formatert *Settings*-dialogen for bedre lesbarhet, og gruppering av *settings*.

Ifølge tidligere arbeider skal *settings* være revidert våren 2013, men vi kan ikke se at alt som er rapportert har blitt gjennomført i versjon **3.0.0**.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmSettings, .Designer, .resx
fap2D.Storing	StoringUtilitites
fap2D.Utilities	GlobalSettings GlobalSettings.Gui GlobalSettings.StructuralModel GlobalSettings.OpenGL GlobalSettings.ComputationalModel
<i>Resources</i>	global.fap

## A.9.3 Nye enheter i *settings*

ID	108
Klassifisering	Endring
Modifisert dato	20.05
Modifisert av	Espen, Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	Juni 2013
Rapportert av	Kristian Pedersen

### Endringer

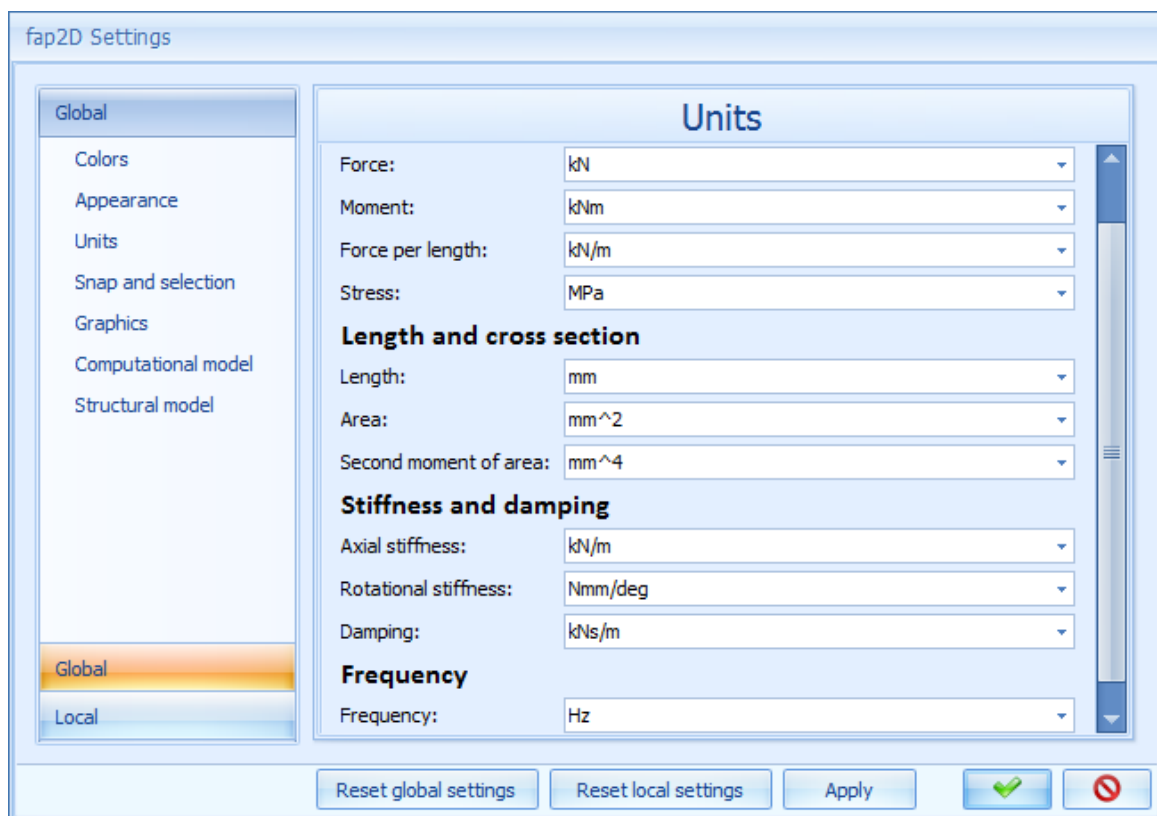
Det er lagt til mulighet for å endre enhetene som brukes for mange av de fysiske størrelsene i **fap2D**, i *settings*.



Tabell A.5: Enheter som nå kan endres i *settings*.

Størrelse	Standard	Lagt til
Frekvens	<b>Hz</b>	rad/s
Rotasjonsstivhet	<b>kNm/deg</b>	Nmm/deg, kNm/rad
Aksialstivhet	<b>kN/m</b>	kN/mm, N/mm
Kraft/lengde	<b>kN/m</b>	N/mm, N/m
Spenninger	<b>N/mm<sup>2</sup></b>	MPa, $N/m^2$
Demping	<b>kNs/m</b>	Ns/mm, Ns/m
Areal	<b>mm<sup>2</sup></b>	$m^2$
2. arealmoment	<b>mm<sup>4</sup></b>	$m^4$

Namespace	Endrede klasser
fap2D.Storing	StoringUtilities.cs GlobalSettingsDataSet.xsd
fap2D.Gui	FrmSettings.cs FrmSettings.Designer.cs

Figur A.146: *Settings* med nye enheter.

## Detaljer

Man har tatt utgangspunkt i et notat om enheter laget av Kolbein Bell [fap2D\_note\_3 vedlegg eller kilde?]. Støtte for enhetene var allerede implementert, men ikke tatt i bruk. Disse er nå lagt til i *Settings* under *Global/Units* som vist i figur A.146.

Det er flere grunner til at det er lagt til så mange nye enheter. For det første slipper brukeren å gjøre omregninger som kan føre til menneskelige feil. Av egen erfaring vet man at slike omregninger blir tatt for lett på. Dette gjelder spesielt frekvens og rotasjonsstivhet, som har kompliserte omregningsfaktorer.

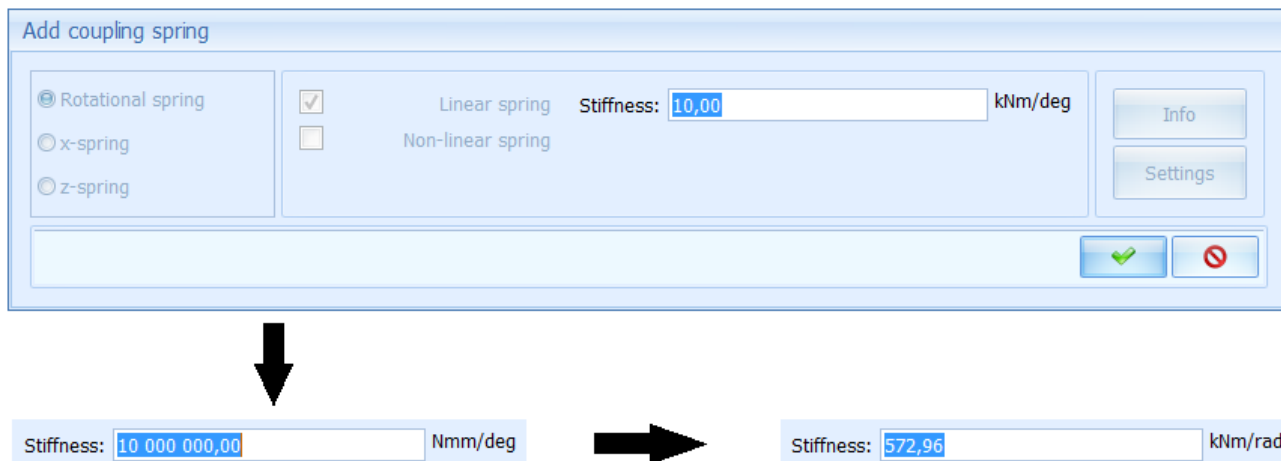
Dessuten var noen av standardenhetene som brukes ikke egnet til å lage små størrelser, slik som liten demping, aksialstivhet og kraft/lengde. Dette problemet henger sammen med at det bare brukes to sifre etter desimaltegn når brukeren angir størrelsen.

En bakside med å innføre flere enheter i *settings* er at enhetene i de fysiske størrelsene bør stemme mest mulig overens. Man bør eksempelvis ikke oppgi krefter i [N] og momenter i [kNm]. Det er altså nå opp til brukeren å gjøre gode valg på dette området.

Følgende eksempler viser korrekt omregning av rotasjonsstivhet og frekvens:

### Rotasjonsstivhet

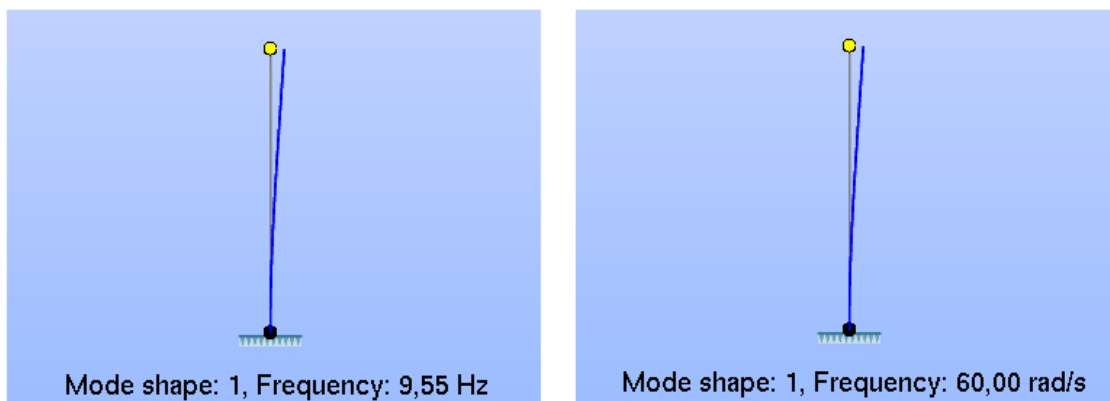
$$\begin{aligned} 10,00 \text{ kNm/deg} &= 10,00(1000 \cdot 1000) \text{ Nmm/deg} = 10000000,00 \text{ Nmm/deg} \\ &= 10,00 \frac{180 \text{ kNm}}{\pi} /_{\text{rad}} \approx 572,96 \text{ kNm/rad} \end{aligned}$$



Figur A.147: Ulike enheter for rotasjonsstivhet.

### Frekvens

$$9,55 \text{ Hz} = 9,55 \cdot 2\pi \text{ rad/s} \approx 60,00 \text{ rad/s}$$



Figur A.148: Ulike enheter for frekvens.

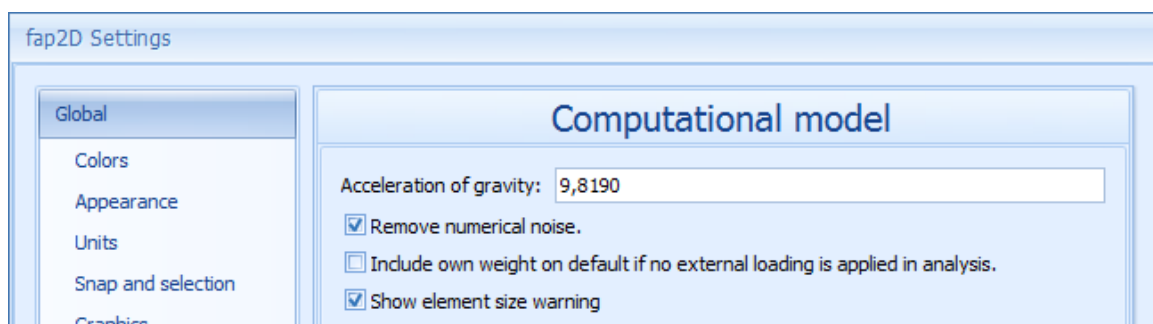
### A.9.4 Advarsler i Settings

ID	109
Klassifisering	Endring
Modifisert dato	21.05
Modifisert av	Espen
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	20.05
Rapportert av	

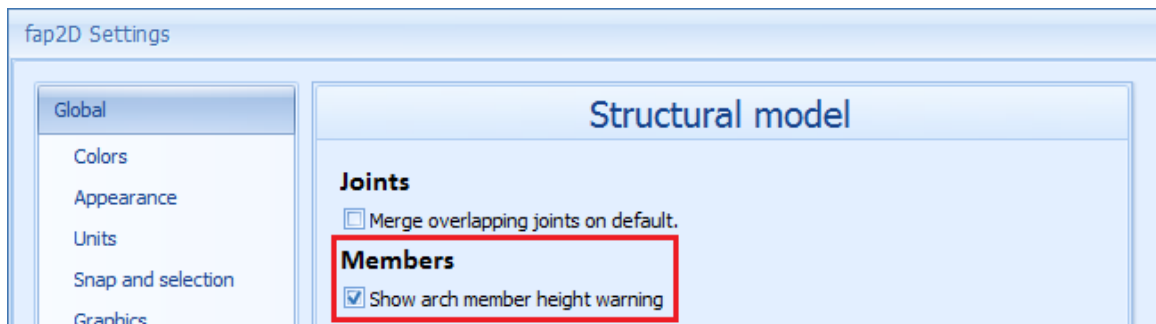
#### Endringer

Lagt til nye innstillinger i *settings*. De nye innstillingene avgjør om advarsler skal vises for små elementer og for høye *arch members* som kan føre til numeriske problemer og modelleringsfeil.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmEccentricity.cs FrmMemberProperties.cs FrmMainGui.AnalysisPage.cs FrmMainGui.LeftPanel.cs FrmModel.cs FrmSettings.cs FrmManageMeshingDialog.cs
fap2D.Utilities	GlobalSettings.cs
fap2D.Storing	StoringUtilities.cs



Figur A.149: Avkryssningsboks i settings, under kategorien *computational model*, for å aktivere/deaktivere advarsler for elementstørrelse.



Figur A.150: Avkryssningsboks i settings, under kategorien *structural model*, for å aktivere/deaktivere advarsler for *arch*-høyde.

## Detaljer

Innstillingene er i form av avkryssningsbokser, og legges under *Global Settings* som gjelder på tvers av konstruksjonsmodeller. Innstillingene påvirker om advarsler, i form av dialogbokser, skal vises dersom numerisk feil kan oppstå.

Som standard er advarsler for både elementstørrelse og *arch* høyde er aktivert.

Det er valgt å kun skrive i klartekst i dialogboksene at de kan fjernes fra i Settings-menyen, istedenfor å inkludere et valg for dette i dialogboksen, for å spare tid.

### A.9.5 *Zoom to model setting*

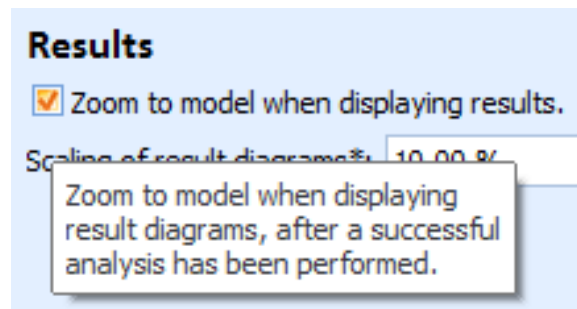
ID	132
Klassifisering	Nytt
Modifisert dato	14.05
Modifisert av	Torjus
Rapportert i versjon	3.5.0 Developer
Rapportert dato	13.05
Rapportert av	Kolbein

## Endringer

Lagt til ny innstilling i *settings*. Innstillingen bestemmer om *zoom to model* skal brukes når en analyse er ferdig kjørt, slik at hele modellen vises i resultatvisningen. Dersom innstillingen er huket bort vil zoom-nivået ikke endres ved analysekjøring.

Innstillgen er brukt i forbindelse med punkt A.1.11.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmSettings
fap2D.Utilities	GlobalSettings
fap2D.Storing	StoringUtilities

Figur A.151: *Zoom to model* i *settings*.

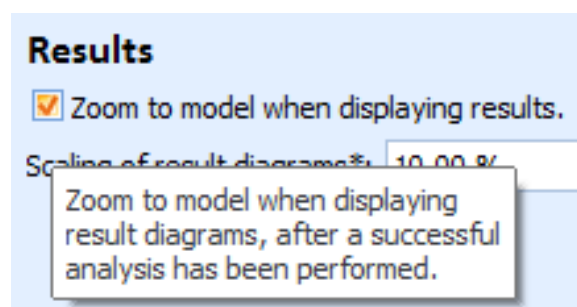
### A.9.6 Mer informasjon om *settings* vha. *tooltips*

ID	133
Klassifisering	Nytt
Modifisert dato	27.05
Modifisert av	Torjus
Rapportert i versjon	3.0.0 Beta
Rapportert dato	Vår 2014
Rapportert av	Torjus

#### Endringer

I forbindelse med at *tooltips* har blitt revidert<sup>6</sup> har det blitt lagt til helt nye *tooltips* for nesten alle punktene i *settings*.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmSettings, .Designer, .resx

Figur A.152: Eksempel på et nytt *tooltip* i *settings*.

#### Detaljer

Tidligere fikk brukeren veldig lite informasjon om hvordan ulike innstillinger fungerte. I praksis virket det som om *settings* ikke var ment å brukes av andre enn utviklerne. Det er gjort kraftig om nå, slik at man faktisk kan bruke dem til noe praktisk.

<sup>6</sup>Se punkt A.11.1 for detaljer.

### A.9.7 Snap og selection settings

ID	134
Klassifisering	Endring, Nytt
Modifisert dato	15.05
Modifisert av	Torjus
Rapportert i versjon	3.3.0 Beta
Rapportert dato	Vår 2013
Rapportert av	Torjus

#### Endringer

Fjernet en innstilling for *snap*:

- *Snap radius (grid)*. Innstillingen er fjernet fordi den ikke er i bruk. Sannsynligvis har hensikten vært å la brukeren bestemme hvor nærmeste hjørnepunktene i rutenettet man må trykke for å *snappe* til dem. I **fap2D** vil det nå alltid *snappes* til *grid* når *snap to grid* er aktivert, så innstillingen er ikke lenger relevant.

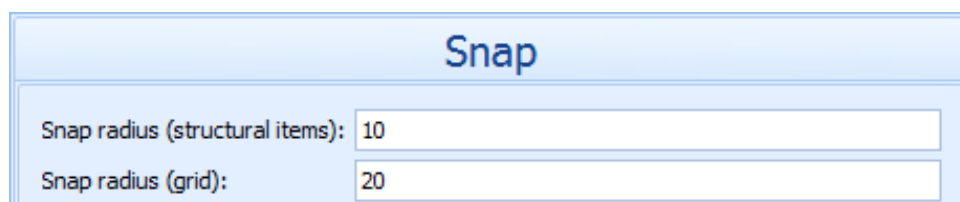
Redigert én innstilling for *selection*:

- *Snap radius (structural items)* har endret navn til *Selection radius (other items)*. Innstillingen blir brukt til å markere *structural items* i *user controls* i utvalgte dialogbokser og for å markere laster. Bruksområdene var de samme i versjon **3.0.0 Beta**. Se figur A.157.

Nye innstillinger for *selection*:

- *Arch member selection offset*. Denne innstillingen angir hvor nær senterlinjen i en *arch member* man må trykke for å markere den. Siden *arch members* er buede har det nemlig ikke vært mulig å bruke samme logikk som for markering av rette *members*. Se punkt A.3.4 (og A.3.5) for mer informasjon. Tallverdien er i piksler.
- *Element selection offset*. Innstillingen angir hvor nær senterlinjen i et element man må klikke for å markere det. Innstillingen gjelder for alle resultatvisninger bortsett fra *steel design*. Se punkt A.3.3, A.7.3, A.3.5 for detaljer. Tallverdien er i piksler.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmSettings
fap2D.Utilities	GlobalSettings
fap2D.Storing	StoringUtilities



Figur A.153: Før endring: *Snap settings*.

**Snap and selection**

Arch member selection offset:

Element selection offset:

Selection radius (other items):

Figur A.154: Etter endring: *Snap og selection settings*.

Arch member selection offset:

Element selection offset:

The selection offset indicates how close to an arch member you must click to mark it as selected.

The default value is 15 pixels. Too low values make it difficult to select arch members. Too high values may make it difficult to select other items.

Figur A.155: Nytt: *Arch member selection offset*.

Element selection offset:

Selection radius (other items):

The selection offset indicates how close to an element you must click to mark it as selected.

The default value is 15 pixels. Too low values make it difficult to select elements. Too high values may make it difficult to select other items.

Figur A.156: Nytt: *Element selection offset*.

Selection radius (other items):

The selection radius indicates how close to a structural item you must click to mark it as selected. Affected items and dialogs:  
Loads, coupling spring dialog, eccentricity dialog, joint dialog, release dialog.

Figur A.157: Etter endring: *Selection radius (other items)*.

## Detaljer

Man kan diskutere nytten av disse innstillingene, men er prioritert fordi utviklerne ikke har brukt mye tid på å optimalisere piksel-verdiene. Standard-verdiene som er satt i *settings* vil fungere i de fleste tilfeller, men kan i noen tilfeller virke dårlig, for eksempel ved å zoome langt ut. Derfor har brukeren mulighet til å manipulere verdiene for å få markert riktig *item*. Dessuten gir det brukeren mulighet til å påvirke markerings-preferansen, som beskrives i punkt A.3.6.

### A.9.8 *Include own weight on default setting*

ID	135
Klassifisering	Nytt
Modifisert dato	12.05
Modifisert av	Torjus
Rapportert i versjon	3.3.0 Developer
Rapportert dato	
Rapportert av	Kolbein, Torjus

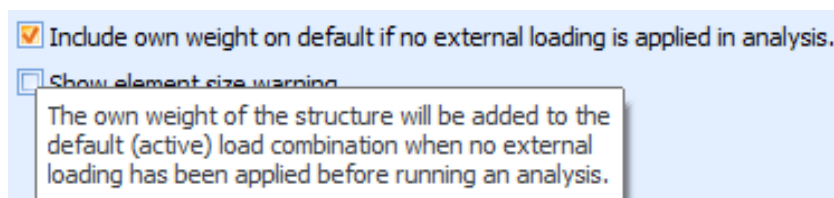
#### Endringer

Lagt til ny innstilling i *settings*. Innstillingen bestemmer om egenvekt skal inkluderes i *default load combination* dersom det ikke er lagt til laster i modellen. Innstillingen gjør seg gjeldende når brukeren trykker på *Run analysis*-knappen for lineær statisk analyse.

Dersom laster legges til er det bare disse som blir å finne i *default load combination*, og egenvekt må da legges til manuelt.

Innstillingen er brukt i forbindelse med punkt A.5.5.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmSettings
fap2D.Utilities	GlobalSettings
fap2D.Storing	StoringUtilities



Figur A.158: *Include own weight on default setting*.

### A.9.9 *Merge overlapping joints setting*

ID	136
Klassifisering	Nytt
Modifisert dato	15.05
Modifisert av	Torjus
Rapportert i versjon	3.3.0 Developer
Rapportert dato	
Rapportert av	Espen

#### Endringer

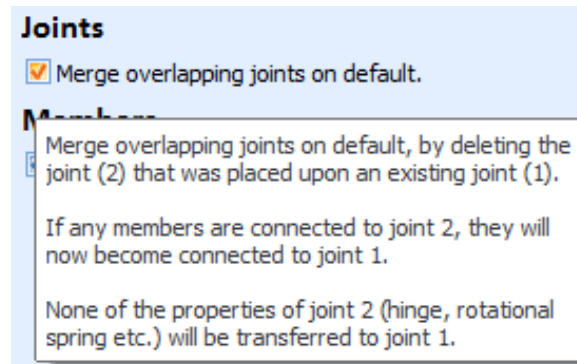
Lagt til ny innstilling i *settings*. Innstillingen bestemmer om *joints* som flyttes slik at de havner over andre *joints* skal slås sammen som standard.



Dersom innstillingen ikke er aktiv vil brukeren få opp en dialogboks når det skapes to overlappende *joints*. I dialogen får man valget mellom å slå *joints* sammen eller beholde begge.

Se punkt A.4.10 for mer informasjon.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmSettings
fap2D.Utilities	GlobalSettings
fap2D.Storing	StoringUtilities



Figur A.159: *Merge overlapping joints setting.*

### A.9.10 Acceleration of gravity setting

ID	102
Klassifisering	Nytt
Modifisert dato	14.05
Modifisert av	Torjus
Rapportert i versjon	3.5.0 Developer
Rapportert dato	
Rapportert av	Torjus

#### Endringer

Lagt til ny innstilling i *settings*. Innstillingen gjør det mulig å sette verdien for gravitasjonsakselerasjonen, **g**.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmSettings
fap2D.Utilities	GlobalSettings
fap2D.Storing	StoringUtilities

Acceleration of gravity: 9,8190

Figur A.160: *Acceleration of gravity setting.*

## A.10 Validering av input

### A.10.1 Formatering og validering av numerisk input

#### Feil

Det oppstår mange *exceptions* i **fap2D** fordi det ikke er tilstrekkelig kontroll på den numeriske inputen som brukeren angir. Noen *exceptions* blir tatt høyde for, men det er også mange som ikke fanges opp (og fører til programstopp).

#### Endringer

Det meste av numerisk input kontrolleres nå, slik at brukeren ikke kan angi ugyldig input. Med ugyldig input mener man alle tegn som ikke kan brukes til å representere tall, slik som bokstaver og spesialtegn. Kontrollene som er innført er disse:

- Formatet på input-tekstfelter (TextEdits) kontrolleres vha. Mask-egenskapene. Mask sørger for at input må skrives på akkurat det formatet utvikleren ønsker. Dvs. at man kan kontrollere hvilke og hvor mange tegn som brukes. Figur A.163 illustrerer bruken.
- Førstnevnte alternativ er ikke brukt for alle input-felter fordi formatet på input tillater visse bokstaver og spesialtegn. Feks. så må det noen ganger være mulig å angi input på standardform, og da må man tillate bruk av bokstaven "E". Disse inputfeltene kontrolleres vha. *parsers* (eller tolkere på norsk) som kan finne ut om input er gyldig etter at det er tastet inn. Eksempel på en *parser test* vises i figur A.162.

#### Argumentasjon

Endringene reduserer faren for programstopp og gjør programmet mer robust.

Den bedrer også arbeidsforholdene for utviklerne fordi strengere kontroll av input gjør at det blir mindre oppryddingsarbeid. Det handler rett og slett om å gjøre gode forberedelser.

Endringene som er nevnt her er forsøkt innført generelt i programmet. Det har ikke blitt tid til å ta for seg alt, for validering er en omfattende prosess. Dessuten har det blitt brukt mange forskjellige metoder for validering i **fap2D** opp gjennom årene. Systematikken er ikke spesielt god.

Punkt A.4.10, A.4.12 og A.4.1 tar for seg validering som er gjennomført for spesielt problematiske områder.

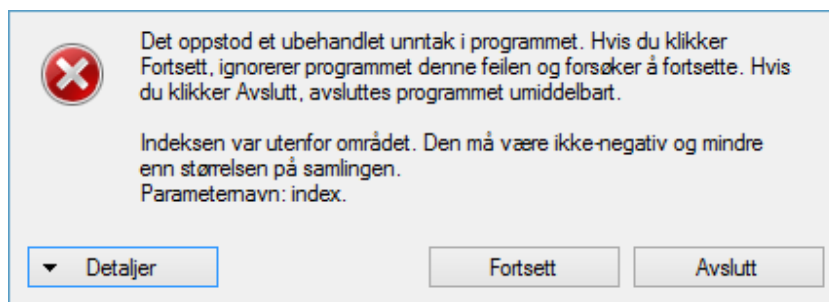
#### Metadata

ID:	43
Klassifisering:	Endring, Feil
Sist modificert:	27. mai
Modifisert av:	Torjus
Rapportert i versjon:	3.0
Rapportert dato:	-
Rapportert av:	-

#### Endrede klasser

Det er såpass mange klasser som har blitt endret at det er poenngløst å liste dem.

#### Figurer



Figur A.161: Poenget med endringene er å unngå *unhandled exceptions* som fører til programstopp.

```
private void txtN_per_Validating(object sender, S
{
    int number;

    if (int.TryParse(txtN_per.Text, out number))
        if (number < 1)
            txtN_per.EditValue = 1;
}
```

Figur A.162: Koden viser bruk av `int.TryParse(...)` som er en metode som sjekker om input kan representeres som et heltall (*integer*). Hvis input ikke kan det, vil det ikke gjennomføres noen endringer. Dermed holder verdier i programmet seg gyldig. Dt finnes tilsvarende metoder for flyttall.

```
// Set display format
u = Units.Instance.GetUnit(UnitType.area);
txtEditA.Properties.Mask.UseMaskAsDisplayFormat = true;
txtEditA.Properties.Mask.MaskType = MaskType.Numeric;
txtEditA.Properties.Mask.EditMask = u.FormatString;
```

Figur A.163: Koden viser hvordan `Mask` properties brukes til å kontrollere formatet i input-tekstfelter (og output-tekstfelter). `MaskType` bestemmer input-typen (numerisk, tekst), `EditMask` bestemmer selve formatet (heltall, desimaltall, ...).

## A.10.2 Kontroll av tekst-input (navn på *load combinations*, *load cases*, ...)

### Metadata

ID	110
Klassifisering	Feil
Modifisert dato	-
Modifisert av	Espen, Torjus
Rapportert i versjon	3.0
Rapportert dato	12.02
Rapportert av	Espen

### Endrede klasser

```
fap2D.Gui FrmLoadCombinationDialog.cs
          FrmLoadTD.cs
          FrmRealTimeLoadTrainDialog.cs
          FrmInfluenceLineLoadTrainDialog.cs
```

## Feil

Det er mulig å gi tomt navn for flere ulike tekst-verdier som var knyttet til objekter som f.eks. *load combinations* og *load cases*. Dette er ikke bare upraktisk, men skaper problemer ved *parsing* av lagrede filer, dvs. når informasjonen i filene skal tolkes i det de lastes inn i programmet igjen.

## Endringer

Endringer er gjort slik at navn i en del tekstfelter ikke kan stå blankt. Et "rødt kryss"-symbol markerer at teksten er ugyldig.

Tekstfeltet som inneholder navnet håndterer nå tilhørende *Validating event* som kalles hver gang tekstfeltet brukes eller bytter fokus. Denne *eventen* kaller metoden `ValidateNameNotNullOrEmpty(...)` i `ValidationRules.cs` som sørger for at navne-objektet eksisterer og ikke er en tom streng.

## Argumentasjon

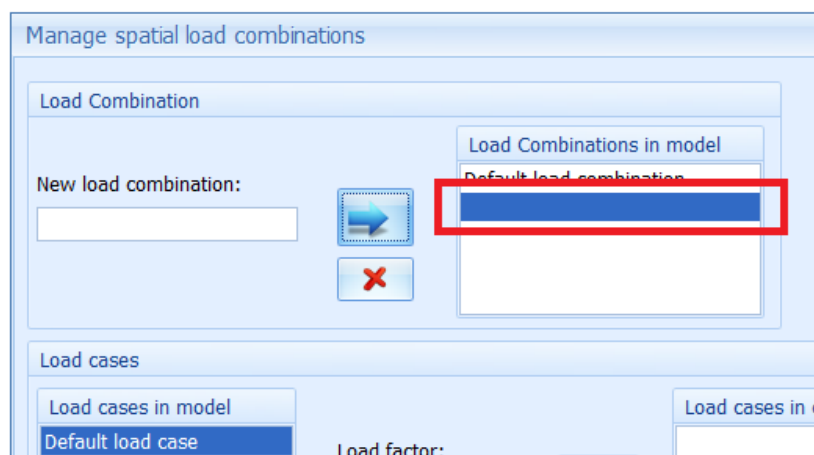
Lastkombinasjoner uten navn førte til at modeller kunne lagres uten at <name>-feltet ble opprettet i lagringsfilen. Dermed kunne ikke den lagrede filen åpnes senere fordi *parseren* kastet en *exception*.

Dette poengterer to ting:

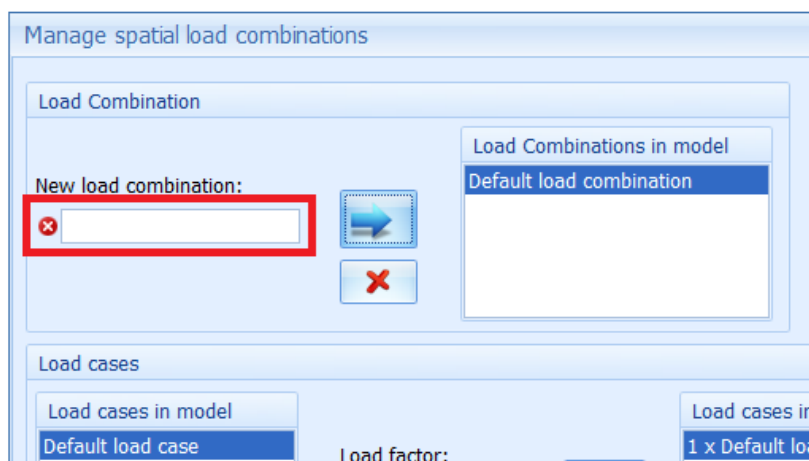
1. Man må gjennomgå og sørge for mer utbredt validering av brukerinntut i **fap2D**.
2. Programmet bør kunne ta hensyn til lagring og lasting av delvis korrupte filer.

Validering av numerisk input blir sett på i punkt A.10.1 Formatering og validering av numerisk input, og lagring blir sett på i punkt A.12.2 Forbedret lagring og lasting av filer.

## Figurer



Figur A.164: Versjon 3.0: En lastkombinasjon er opprettet uten navn.



Figur A.165: Versjon 3.1: Lastkombinasjonen opprettes ikke hvis feltet for navn er tomt.

## A.11 Informasjon til brukeren

### A.11.1 *Tooltips og super tooltips*

ID	6
Klassifisering	Endring, Nytt
Modifisert dato	27.05
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	November 2013
Rapportert av	Torjus

#### Endringer

*Tooltips* og *super tooltips* har blitt revidert, både når det gjelder innhold/tekst og formatering. De nye *tooltips*-ene er ryddigere og en del av dem er skrevet om for å bli mer presise.

Referanser til aktuelle avsnitter i *User's Manual* oppgis der det er naturlig at brukeren vil søke mer informasjon.

Det er lagt til informasjon om hurtigtaster der det ikke har vært det før, for eksempel "CTRL-O" for å åpne en modellfil.

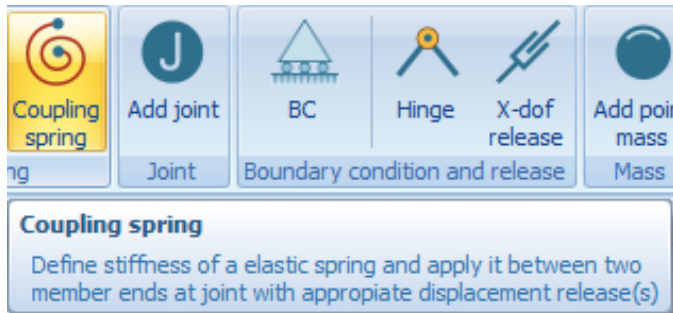
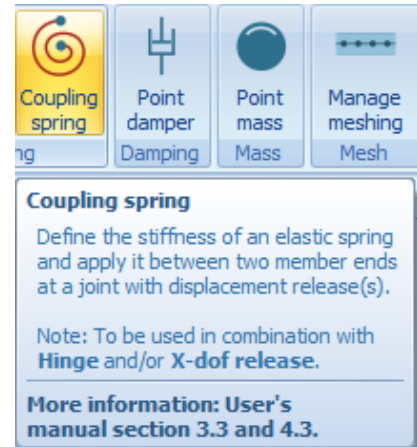
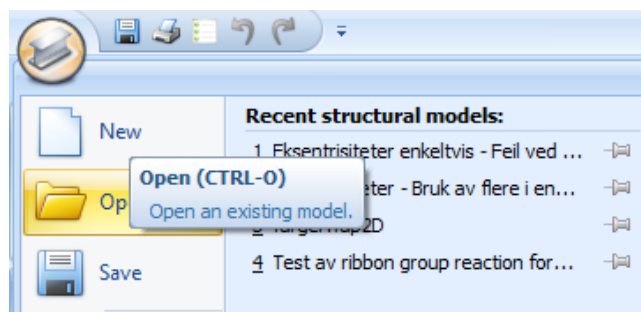
*Tooltips* og *super tooltips* er lagt til for *controls* som ikke har det fra før av, blant annet *Draw member*. Dessuten har de fleste punkter i *settings* fått *tooltip*.

En fullstendig oversikt over alle endringene finnes i vedlegg B.2.6 Konklusjon.

Følgende deler av brukergrensesnittet har blitt revidert:

*Application menu*, *quick access toolbar*, *toolbox panel*, *status bar*, *modelling ribbon page*, *loading ribbon page*, *analysis ribbon page*, *results ribbon page* og *settings*-dialogen.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmMainGui.Designer, .resx FrmExportDialog.Designer, .resx FrmImportDialog.Designer, .resx FrmSettings.Designer, .resx FrmXtrmDialog.Designer, .resx FrmUtilizationDialog.Designer, .resx

Figur A.166: Før endring av *super tooltip*.Figur A.167: Etter endring av *super tooltip*.Figur A.168: Etter endring: Eksempel på nytt *tooltip*, med hurtigtast.

## Detaljer

For detaljer refereres det til vedlegg B.2.6. Endringene er for omfattende til å presenteres her.

### A.11.2 Formatering av *information list*

ID	37
Klassifisering	Endring
Modifisert dato	03.02
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	02.02
Rapportert av	Torjus

## Endringer

*Information List* er blitt formatert for å gjøre informasjonen om analyser mer oversiktlig, og det er lagt til et tidstempel. Lesbarheten i listen er forbedret, men den kunne vært forbedret ytterligere, med mer "luft".

1. Konsekvent bruk av kolon, ikke parenteser.
2. Konsekvent bruk av stor/liten forbokstav.
3. Blank linje mellom analyser.
4. Tidsstempel som viser når analysen ble fullført.

<i>Namespace</i>	Endrede klasser
fap2D.Computations	ComputationalModel.ModelTools ComputationalModel.Analyses

```

Information
Mesh Generated (milliseconds used: 18,001)
Nodes generated: 51
Elements generated: 50
Time used to create tables in milliseconds: 23,0013
-----
Starting Linear Static Analysis...
Linear Static Analysis Finished.
-----
Time used by Frame2D_LS in milliseconds: 20,0012
Mesh Generated (milliseconds used: 0)
Nodes generated: 51
Elements generated: 50
Time used to create tables in milliseconds: 0
-----
Starting Free Vibration Analysis...
Number of requested eigenvalues: 5
Number of accepted eigenvalues: 5
f_1: 24,7754489861933
f_2: 61,2143242128211
f_3: 112,500876114735
f_4: 130,58464591876
f_5: 179,033632893861
Free Vibration Analysis Finished.
-----
Time used by Frame2D_FV in milliseconds: 54,0031

```

Figur A.169: Før endring.

```

Information
Time used to generate mesh in milliseconds: 15,0008
Nodes generated: 51
Elements generated: 50
Time used to create tables in milliseconds: 17,001
-----
Starting Linear static analysis...
Linear static analysis finished at: 14:34:04
-----
Time used by Frame2D_LS in milliseconds: 9,0005
Time used to generate mesh in milliseconds: 0
Nodes generated: 51
Elements generated: 50
Time used to create tables in milliseconds: 0
-----
Starting Free vibration analysis...
Number of requested eigenvalues: 5
Number of accepted eigenvalues: 5
f_1: 24,7754489861933
f_2: 61,2143242128211
f_3: 112,500876114735
f_4: 130,58464591876
f_5: 179,033632893861
Free vibration analysis finished at: 14:34:10
-----
Time used by Frame2D_FV in milliseconds: 59,0034

```

Figur A.170: Etter endring.

### A.11.3 Tidsstempel i *error list* og (*information list*)

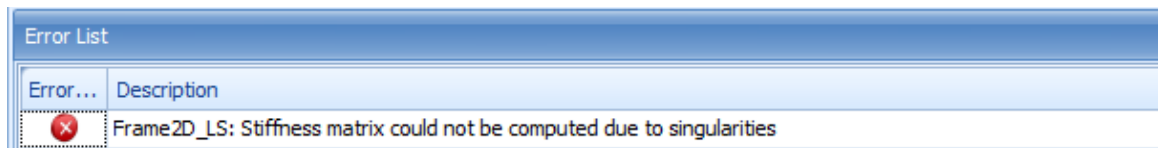
ID	44
Klassifisering	Nytt
Modifisert dato	22.01
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	November 2013
Rapportert av	Torjus

## Endringer

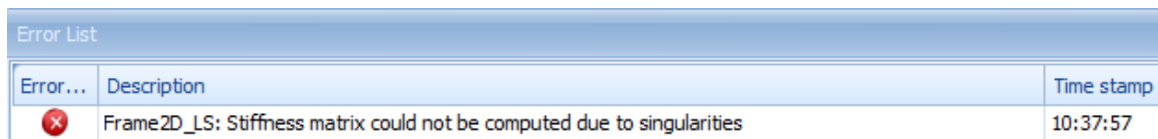
Tidsstempel er lagt til for alle feilmeldinger slik at det er enklere å skille feilmeldingene fra hverandre i *error list*. Det vises ofte flere feilmeldinger samtidig når noe går galt, så et tidsstempel gjør det enklere å se hvilke som henger sammen.

Et tidsstempel, som viser tidspunktet da analyser, er ferdige er lagt til i *information list*. Dette er kanskje ikke like nyttig som for feilmeldinger, men det kan godt hende noen setter pris på det.

<i>Namespace</i>	Endrede klasser
fap2D.Utilities	Error ErrorList Alle klasser som oppretter nye <i>errors</i>
fap2D.Computations	ComputationalModel.ModelTools ComputationalModel.Analyses



Figur A.171: Før endring.



Figur A.172: Etter endring.

### A.11.4 Lese og kopiere feilmeldinger i *error list*

ID	45
Klassifisering	Nytt
Modifisert dato	03.02
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	02.02
Rapportert av	Torjus

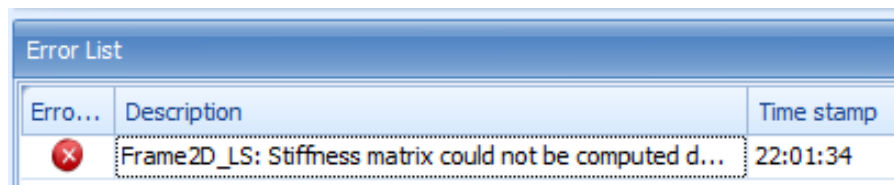
## Endringer

Kolonnebredden i *error list* er nå justerbar slik at man kan få sett hele teksten i en feilmelding.

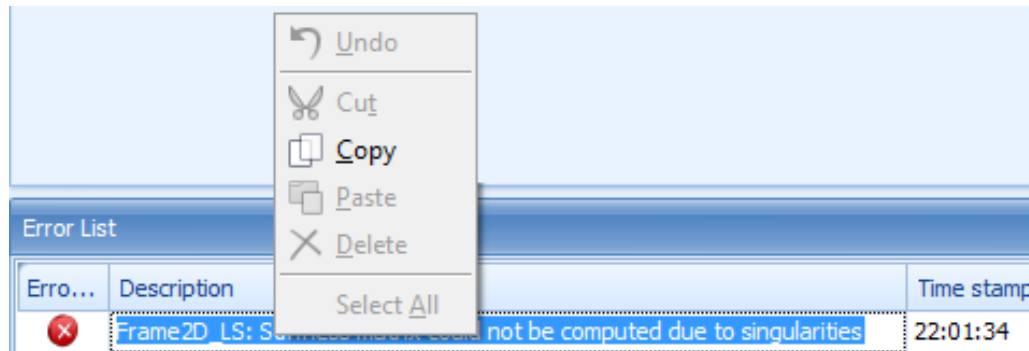
Teksten i hver feilmelding kan nå kopieres. Praktisk dersom brukere trenger hjelp fra utviklerne, så slipper de å skrive hele feilmeldingen om igjen.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmMainGui.BottomPanel





Figur A.173: Før endring.



Figur A.174: Etter endring. Kolonnen justeres slik at all tekst vises, og det er mulig å kopiere.

### A.11.5 Musepeker-koordinater i *status bar*

#### Endringer

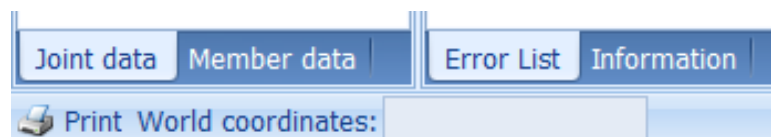
Innføring av felt for visning av musepeker-koordinater. Feltet er endret til en *label* som skriver ut musepekerens posisjon i samme koordinatsystem som modellen.

#### Metadata

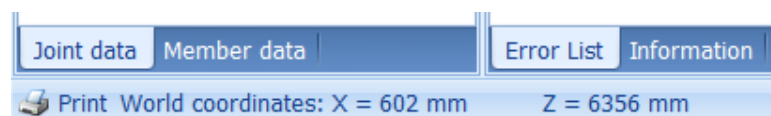
ID:	29
Klassifisering:	Endring
Sist modifisert:	16.05
Modifisert av:	Espen
Rapportert i versjon:	3.0
Rapportert dato:	Høst 2013
Rapportert av:	Espen

#### Endrede klasser

```
fap2D.Gui FrmMainGui.cs
          FrmModel.cs
```



Figur A.175: Før endring.



Figur A.176: Etter endring.

## Detaljer

I programmets *status bar* har det vært et tekstfelt med en tilhørende tekst: "*world coordinates*"<sup>7</sup>. Denne hadde ingen funksjon, men var tidligere tiltenkt å skulle vise koordinatene til musepekeren i koordinatsystemet.

Tekstfeltet er endret til en *label* (tekst som ikke kan markeres eller endres av bruker) som skriver ut musepekerens posisjon i samme koordinatsystem som modellen.

Teksten som skrives ut er:

```
"World coordinates: X = # mm           Z = # mm"
```

der "#" er pekerens posisjon rundet til nærmeste heltall.

Posisjoner trenger kun heltalls-presisjon fordi de skal brukes i tandem med *snap*-funksjonen. Dersom presisjon er nødvendig bør *JointData*-panelet benyttes for å skrive inn koordinater manuelt.

Enhetene koordinatene vises i er det samme som lengdeenheter vises i resten av programmet. I skrivende stund er dette begrenset til meter [m] og millimeter [mm].

Det er valgt å skille koordinatene med kun mellomrom (10 stk.) for å enkelt skille mellom X- og Z-koordinatene. Alternativt kunne koordinatene vises som "(#, #)" men da må enhetene plasseres et annet sted.

### A.11.6 Tusen-skilletegn og prosentvisning

ID	131
Klassifisering	Endring
Modifisert dato	04.04
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	Høst 2013
Rapportert av	Torjus

## Endringer

"Tusen"-skilletegn er innført for mange input- og output-verdier slik at det er lettere å lese hvor stor en tallverdi er. Rettingen er gjort ved å bruke det numeriske *display*-formatet "n". Derfor knyttes den endringen til XXX ref numeriske tallverdier-validering.



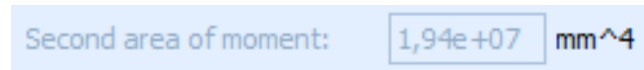
Figur A.177: Før endring t.v., etter endring t.h.

Namespace	Endrede klasser
fap2D.Utilities	Units.Created
	De klassene som bruker enhetene i Units.Created

<sup>7</sup>"World coordinates" refererer til programmets interne navn på koordinatsystemet modellen eksisterer i – alternativet er "Screen coordinates"

## Detaljer

Det er dessverre ikke alle input- og output-verdier som har fått formatet “n”. Det er nemlig andre hensyn som må tas. I tekstfeltet som viser 2. arealmoment for et tverrsnitt er det f.eks. viktig at hele tallet vises i output-feltet. Derfor må formatet “g” brukes for å ta høyde for store tallverdier. Se figur A.178 for en illustrasjon.



Second area of moment: 1,94e+07 mm<sup>4</sup>

Figur A.178: 2. arealmoment vises på kompakt format, “g”.

## A.12 Filbehandling

### A.12.1 Fjerning av slettede modellfiler fra *application menu*

ID	10
Klassifisering	Feil
Modifisert dato	14.02
Modifisert av	Torjus
Rapportert i versjon	3.0.0 - Beta
Rapportert dato	Oktober 2013
Rapportert av	Torjus

### Feil

En modellfil som er slettet eller flyttes fra sin opprinnelige lagringsmappe blir ikke slettet fra listen *Recent structural models* i *Application menu*.

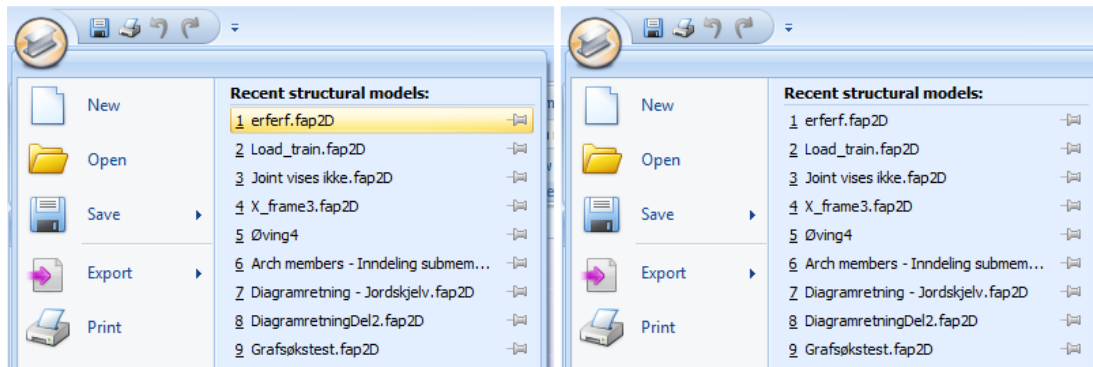
### Endringer

Dersom man prøver å åpne en modellfil som er slettet eller flyttet, i listen over *Recent structural models*, vil filen slettes fra listen. Brukeren får beskjed om at filen ikke eksisterer i den gitte lagringsmappen (implementert fra før).

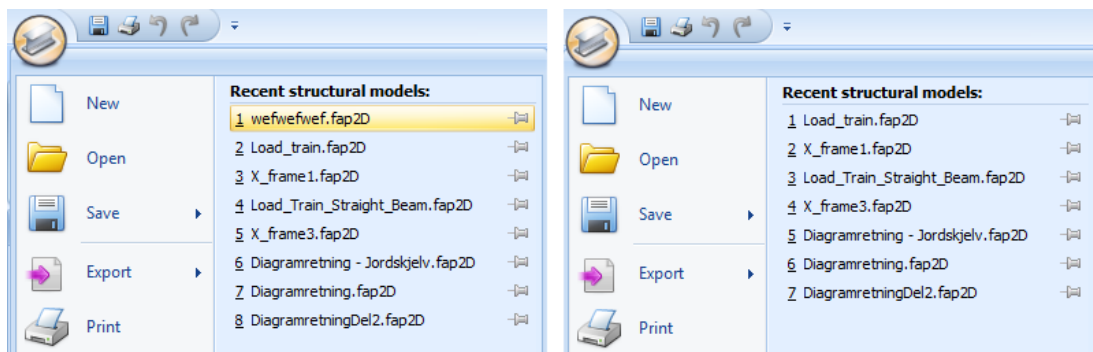
Feilen håndteres vha. de nye metodene `RemoveFromMostRecentFiles(...)` og `RemoveElement(...)`.

**NB:** Sent i masteroppgaven oppdaget man at slettingen ikke fungerer i alle tilfeller. Den fungerer bare når filene slettes eller flyttes fra sin lagringsmappe. Dersom filstien endres ved å for eksempel endre navnet på lagringsmappen vil man ikke klare å slette den! Man har ikke fått tid til å endre dette.

<i>Namespace</i>	Endrede klasser
fap2D.Gui	FrmMainGui.ApplicationMenu FrmMainGui.RecentFiles



Figur A.179: Før endring: Filen som er øverst i listen t.v. burde vært slettet i listen t.h.



Figur A.180: Etter endring: Filen som er øverst i listen t.v. blir korrekt slettet fra listen t.h.

## Detaljer

Denne feilen er rettet opp fordi den kan irritere brukeren. Dersom alle filene i listen er slettede modeller vil man ikke få utnyttet *Recent structural models* lenger, så det var essensielt å rette feilen. Den var relativt enkel å rette så den ble prioritert tidlig som en oppvarming til mer avanserte oppgaver.

Feilen er som flere andre feil med på å gi brukeren en følelse av å ikke være i kontrollsetet.

## A.12.2 Forbedret lagring og lasting av filer

### Feil

Lagring av konstruksjonsmodeller fungerte ikke alltid som den skulle. Lagringen kunne bli avbrutt av feilmelding, noe som krasjet programmet, eller filen kunne bli korrupt slik at den ikke kunne leses inn. Det var også mulig at filer fra ulike versjoner av programmet ikke kunne åpnes eller krasjet programmet.

### Metadata

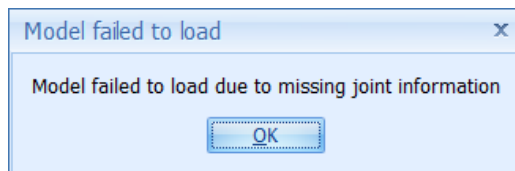
ID:	48
Klassifisering:	Feil
Sist modifisert:	21.02
Modifisert av:	Espen
Rapportert i versjon:	3.0
Rapportert dato:	Januar 2014
Rapportert av:	Brukere

### Endrede klasser

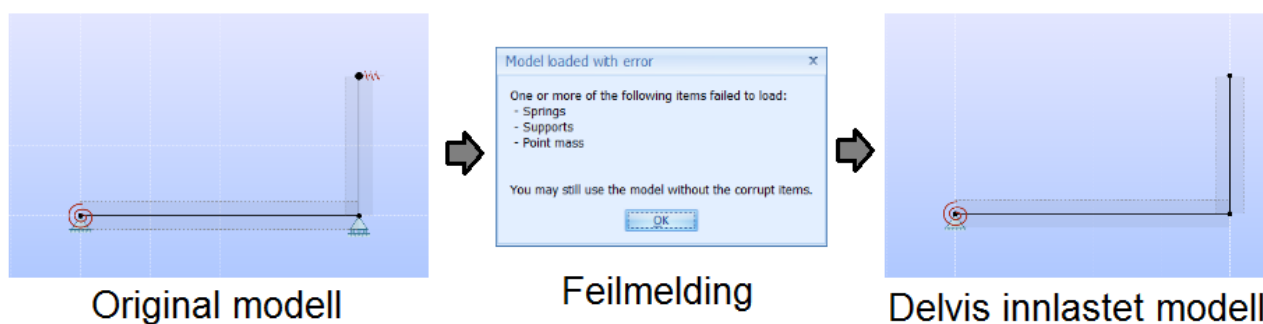
fap2D.Storing	StoringUtilities
fap2D.Gui	FrmMainGui.ApplicationMenu

## Endringer

Det er endret lagrings- og innlastingsmetodene slik at de takler feil i modellen eller filen bedre. *Joints* og *members* er essensielle byggestener for en modell, og må derfor være i orden for å godta lagring. Alle andre *structural items* kan ignoreres dersom de inneholder feil. Ignorerte feil vil bli rapportert til bruker.



Figur A.181: Etter endring - varsel om komplett feil.



Figur A.182: Etter endring - varsel om delvis feil.

## Detaljer

Flere brukere opplevde feil ved lagring og lastning av filer der programmet stopper opp og ingenting blir lagret. Å kunne lagre en fil er essensielt slik at man ikke mister arbeid selv om programmet stopper.

### Beskrivelse

For at lagring skal gjennomføres må hele modellen ha gyldige verdier. Dersom dette ikke er tilfelle vil programmet kaste *exceptions*. Dvs. programmet lukker seg for bruker. Dette kan også skje ved innlasting av en modell selv om den ble lagret uten problemer.

En av hovedårsakene til *exceptions* var `DB_NULL` verdier. Klassen `StoringUtilities.cs` benytter seg av **ADO.NET** rammeverket som i **fap2D** betyr at det eksisterer et databaseskjema med enkle databaserelasjoner. Skjemaet viser hvilke felter som ikke godtas som null-felter, dvs. feltene skal ha verdier for å gi en gyldig database. Om databasen tillater `DB_NULL` betegnes med egenskapen "AllowDBNull" som kan settes til *true* eller *false*. Egenskapen `NullValue = (throw exception)` håndterer hva som skjer når parseren møter en nullverdi. Merk at **MSDN** for `DBNull`-klassen [Microsoft Corporation. *MSDN, DBNull*. <http://msdn.microsoft.com/en-us/library/system.dbnull%28v=vs.110%29.aspx/>] sier at et `DBNull`-objekt ikke er det samme som et objekt-orientert null. Et objekt-orientert null betyr at det mangler referanse til et objekt. `DBNull` kan være en tom databasekolonne eller en ikke-initialisert variant.

## Diskusjon/Løsning

Nullverdier som lagres eller lastes må altså enten sjekkes for og håndteres individuelt, eller så må *exceptions* som kastes fanges opp og håndteres.

Følgende alternativer bør vurderes:

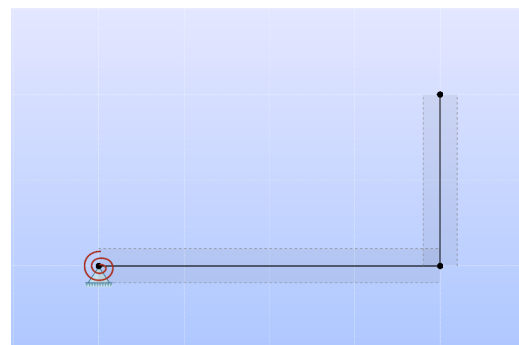
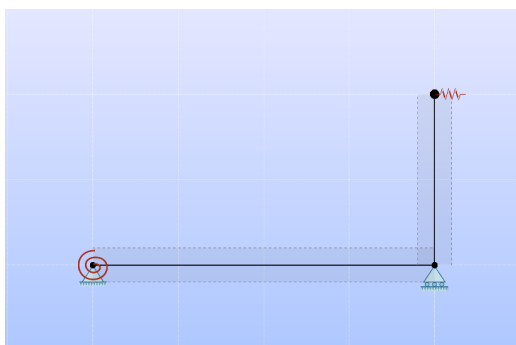
- Det enkleste er å ganske enkelt ikke la bruker lagre/laste hvis det er en feil. Det er ikke gunstig siden bruker potensielt aldri kan fikse feilen.
- Alternativt kan man ikke la bruker lagre/laste, men rapportere om hver feil. Det er arbeidskrevende for utvikler, men gunstig for bruker som da får en gyldig lagret/lastet fil. Det må da vurderes hvor detaljert hver rapport skal være, det diskuteres nedenfor.
- Man kan lagre/laste, men rapportere hva som gikk galt. Man må fremdeles vurdere detaljnivå i rapporteringen. Man risikerer ugyldig lagret/lastet fil, men bruker kan avslutte/åpne programmet med noe som er lagret.
- Det beste er å sørge for at alt lagres/laste gyldig første gang, men det krever spesielt mye arbeid fra utviklere og er vanskelig å garantere.

Å håndtere hver enkelt verdi for hvert *structural item* individuelt er arbeidskrevende og uoversiktlig. Det kan raskt bli mange feil per *structural item*, og mange ting å enten rapportere eller håndtere.

Å håndtere hvert enkelt *structural item* er mindre arbeidskrevende, men avhengigheter mellom ulike *structural items* vil skape problemer dersom hvert problem ikke håndteres skikkelig.

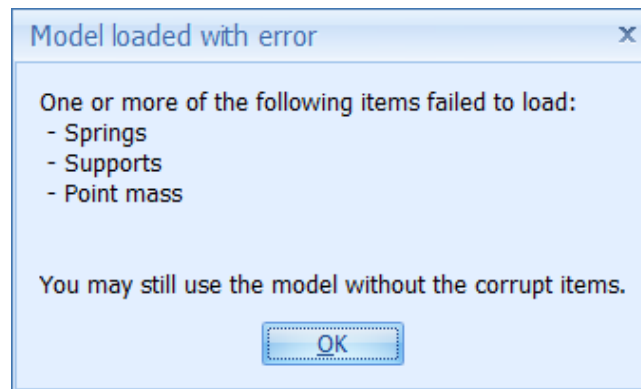
Løsningen som er greiest for alle parter er å sørge for at alt essensielt alltid er OK ved lagring, la bruker lagre, rapportere om hvilke deler som er feil slik at bruker kan fikse det om ønskelig, og sørge for at programmet kan laste manglende deler.

Slik vi har valgt å gjøre det må *joints* og *members* gå gjennom lagre- og lastemetodene uten problemer for at noe skal bli lagret. Hvis noe går galt avbrytes innlastingen og man får opp en dialogboks som vist i figur A.181. Dersom noe annet går galt med andre *structural items* vil det vises i en dialogboks hva som gikk galt, men det som fungerte skal likevel lagres/lastes. Nedenfor er eksempelet i figur A.182 beskrevet der en modell, vist i figur A.183, med *joints*, *members*, opplager, punktmasse og fjær skal lastes inn.



Figur A.183: Original konstruksjonsmodell som Figur A.184: Konstruksjonsmodellen i figur A.183 blir lagret med feil. delvis lastet inn.

Ett opplager, en punktmasse og en fjær har feil i den lagrede filen, og blir ikke lastet. Brukeren vil bli varslet om dette med en dialog som vist i figur A.185, og man får opp en modell uten *structural items* som feilet, se figur A.184. Merk at selv om en type *structural item* feiler betyr det ikke at alle instansene av den typen forsvinner. Selv om både fjær og opplager forsvant finnes det fremdeles en fjær og et opplager i modellen.



Figur A.185: Dialogboks som viser at modellen som ble lastet mangler enkelte *structural items*.

### A.12.3 Fjerne velkomstskjerm fra eksisterende modeller

#### Feil

Når man åpner **fap2D** fra en assosiert fil (en fil med fil-etternavnet .fap2D) åpnes programmet med den vanlige velkomstskjermen istedenfor å åpne den spesifikke filen man klikket på.

#### Endring

Årsaken var at metodene som initialiserer programmet utfører oppgavene i feil rekkefølge. Programmet blir bedt om å åpne den gitte filen før det har startet ordentlig.

For å fikse feilen måtte det legges ved en variabel `_filepath` som holder fil-stien sendt til konstruktøren av `FrmMainGui`, og som senere, i `OnCreateControl()`, åpner filen med metoden:

```
...
if (_filePath != null)
OpenModel(_filePath);
...
```

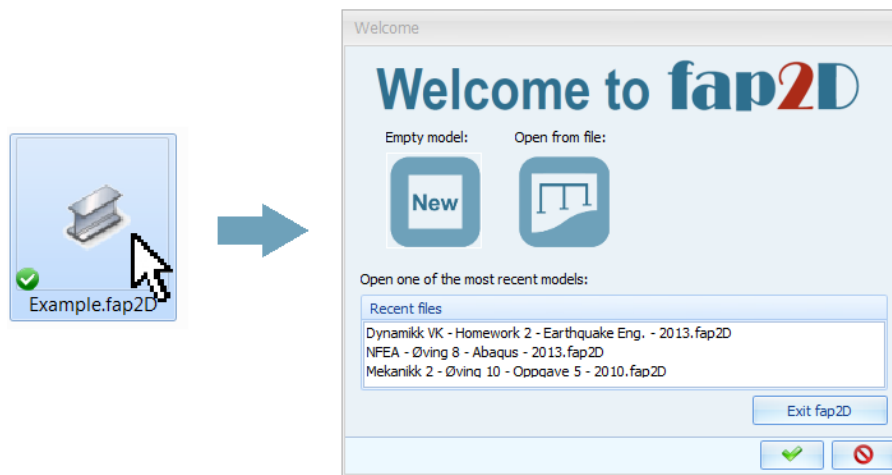
#### Figurer

#### Metadata

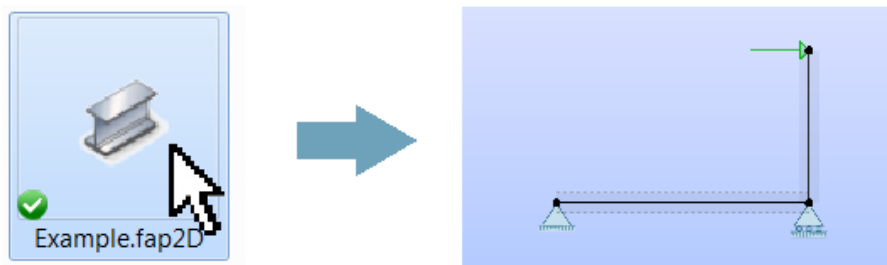
ID:	11
Klassifisering:	Feil
Sist modifisert:	07.05
Modifisert av:	Espen
Rapportert i versjon:	3.0
Rapportert dato:	Høst 2013
Rapportert av:	Espen

#### Endrede klasser

fap2D.Gui FrmMainGui.cs



Figur A.186: Før - fil åpner programmet og viser startskjerm.



Figur A.187: Etter - fil åpner programmet og modellen vises.



# Tillegg B

## Betatesting av fap2D 3.0.0

### B.1 Introduksjon til betatesting

Betatesting er en metode for å teste programvare som innebærer at en større gruppe med personer utenom utviklingsteamet tester om programvaren fungerer i tilstrekkelig grad.

Hovedmålet med testingen er å finne ut om en brukere (kunder) er fornøyde med programmets hovedfunksjoner. Et sekundærmål for utviklerne er å utnytte at mange personer sannsynligvis vil kunne teste robustheten til programmet i mye høyere grad enn et lite utviklerteam. Dessuten kan eksterne personer bidra med nye impulser.

En viktig bestanddel i betatesting er et rapporteringssystem for feil og annen tilbakemelding som utviklerne kan bruke til å forbedre programmet før det slippes en ordinær versjon. I tillegg til et rapporteringssystem er det lurt at utviklerne stiller opp som *support/støttesystem* for brukere som trenger hjelp og har spørsmål. Dette sikrer god toveis kommunikasjon der også brukerne får respons.

### B.2 Gjennomføring av betatesting

Betatestingen er gjennomført for versjon **3.0.0 Beta** tidsperioden 17. januar t.o.m. 23. april 2014. Testingen har foregått i forbindelse med at faglærer Arne Aalberg ønsket å benytte **fap2D** i øvingsopplegget i faget TKT4180 Konstruksjonsmekanikk - Beregningsmetoder (KMEK). Totalt skulle det gjennomføres åtte øvingsoppgaver vha. konstruksjonsprogramvare. Brukerne hadde mulighet til å benytte et alternativt program, Focus Konstruksjon 2D [12], men ble oppfordret til å bruke **fap2D**. Grunnen til at faglærer ville prøve ut **fap2D** er at det er bedre egnet til enkel konstruksjonsanalyse enn Focus Konstruksjon. Sistnevnte er mer rettet mot dimensjonering i hht. Norsk standard.

Betatestingen ble sett på som en gylden anledning til å få tilbakemeldinger på programmet. Både i form av om programmet lar seg installere og kjøre på ulike plattformer, og i form av feilmeldinger og mangler. For å få tilbakemelding gjennomførte studentene frivillig feilrapportering og evaluering av **fap2D**. Vi utførte support og besvarte spørsmål som kom inn på mail.

#### B.2.1 Testgruppe

Testgruppen var studenter som tok faget KMEK våren 2014, ved KT. De fleste av deltagerne var studenter fra 3. årstrinn med fordypning i konstruksjonsteknikk. Studentene har derfor forutsetninger for å forstå de oppgavene og løsningene, som programmet tar for seg.

De har derimot ikke forutsetning for å forstå all den numeriske teorien som løsningsmetodene i **fap2D** benytter seg av, ettersom den læres i løpet av faget. De færreste har brukt lignende programmer tidligere og regnes derfor som nybegynnere. Derfor passer de godt som testgruppe, for målgruppen til **fap2D** er ferske brukere av konstruksjonsprogramvare.

### B.2.2 Demo v/veileder

Veileder Bell tok på seg å holde en demo av **fap2D** for studentene testgruppen slik at de skulle komme raskt i gang. De viktigste stegene for å gjennomføre en analyse ble gjennomgått og det ble advart om tilstanden til programmet. En del av testgruppen hadde begynt å bruke programmet før demoen i og med at øvingsopplegget allerede var i gang. Majoriteten av testgruppen deltok på demonstrasjonen. Etter at veileder var ferdig med demoen, ble det informert om skjema for feilrapportering, evaluering og *support*.

### B.2.3 Support/støtte og feilrapportering

For å motivere til bruk av **fap2D** i øvingsopplegget tok vi på oss å *supportere* studentene ved å svare på mail når problemer oppstod. Det ble i tillegg lagt ut et feilrapporteringsskjema på fagets hjemmeside. Til sammen registrerte vi om lag 20 henvendelser, de fleste i løpet av februar.

Siden **fap2D** er Windows-basert ble det installert en kopi på de fleste PC-ene på datasalene Zevs og Big Ben på Lerkendalsbygget, slik at Mac- og Linux-brukere fikk mulighet til å arbeide med **fap2D**. Det oppstod et problem med skriverrettigheter for studentbrukere på disse datasalene. Ingen fikk utført analyser fordi loggfilen til **fap2D** ble forsøkt skrevet ut til programfil-mappen (*C:/Programfiler/NTNU/Frame Analysis Program/*), men uten skriverrettigheter stoppet programmet. Feilen ble rettet opp i løpet av noen dager med hjelp fra overingeniør Kenneth Sundli.

### B.2.4 Områder av programmet som ble testet

#### Robusthet og plattformuavhengighet

Installasjon og bruk av programmet på ulike plattformer. Ble testet på Windows 7, 8 og 8.1, med både 32- og 64-bits prosessorer. I tillegg til installasjon av **fap2D** fikk vi også testet om tilleggsfilene som trengs fungerte, blant annet Microsoft Office Access Database Engine. Sist, men ikke minst fikk vi testet hvor ofte programmet stanse pga. *unhandled exceptions*.

#### Modellering av konstruksjoner og laster

Hovedfokuset i betatestingen var å teste hvor enkel og feilfri modelleringen av konstruksjoner foregår i programmet. I øvingsoppgavene i *KMEK* ble følgende *structural items* benyttet:

- Bjelker, søyler og staver: *Beams, bars, cables, struts*.
- Leddforbindinger: *Joints, hinges, rotational coupling springs*.
- Fritt opplagte og fast innspente randbetingelser.
- Tverrsnitt og materiale for bjelker, søyler og staver.
- Punkt- og linjelaster.

#### Analysetyper

Testgruppen har forutsetning for å gjennomføre følgende analyser:

- Linear static
- Linearized buckling
- Nonlinear static
- Influence lines (avansert)
- Free vibration (avansert)

Av de nevnte analysene har de tre førstnevnte blitt testet gjennom bruk av **fap2D** i øvingsopplegget i KMEK. Skjærdeformasjoner er blitt inkludert i flere av analysene.

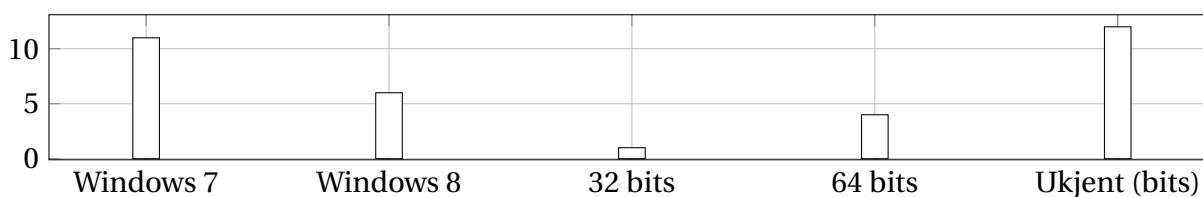
## B.2.5 Evaluering

Evalueringen besto av generelle spørsmål om bruken av **fap2D** og spesifikke spørsmål som ble brukt for å vurdere eventuelle modifikasjoner og nye implementasjoner i programmet. De generelle spørsmålene skulle gi oss et overordnet inntrykk av om programmet er dugende.

### B.2.5.1 Resultater fra evaluering

17 evalueringsskjemaer ble levert inn. Svarene er presentert på de neste sidene.

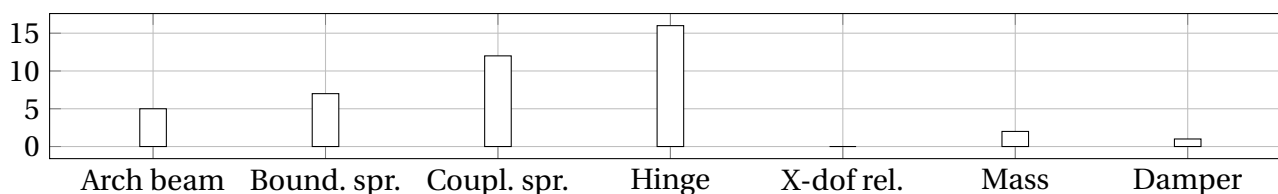
#### *-Hvilket operativsystem ble fap2D installert på?*



#### *-Hvordan gikk installasjonen av fap2D?*

Alle som leverte inn evalueringsskjema fikk installert programmet greit, så lenge man fulgte installasjonsprosedyren som ble gitt. Én student henvendte seg på mail fordi **fap2D** det ikke var mulig å få kjøre pga. feil med Access Database Engine. Det viste seg at ulike PC-er krever ulike versjoner av Access Database Engine (2007 eller 2010), uavhengig av valgt Windows-versjon og prosessorkjerne. Derfor er både 2007- og 2010-versjon lagt til i mappen med installasjonsfiler.

#### *-Har du brukt noen av følgende structural items i fap2D?*



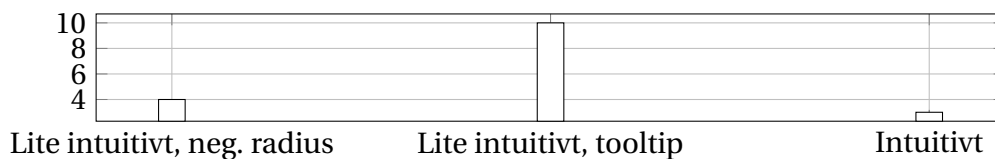
#### *-Var det vanskelig å forstå hvordan noen av structural items skulle brukes?*

Hensikten med spørsmålet var å sjekke om *tooltips*-ene for knappene til *items* var forståelige. I tillegg fikk man tilbakemelding på om *items* fungerte slik de var ment. Noen nevnte at

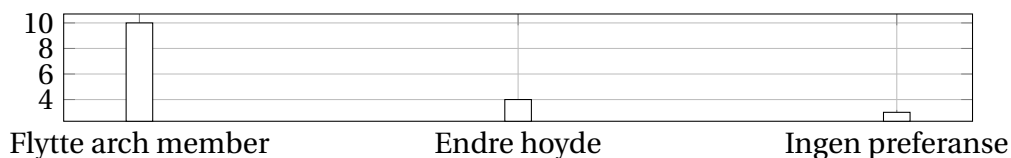
dialogboksen for *coupling spring* var forvirrende. Det var også problemer med å få lagt inn korrekte leddforbindelser og å kombinere ledd med rotasjonsfjærer. De få som prøvde å bruke *arch members* fikk akkurat som oss erfare at programmet stanset. De fleste fikk gjennomført oppgavene de skulle, men måtte gjerne restarte programmet eller starte modelleringen på nytt en gang eller to.

#### **-Lage arch members som krummer nedover:**

I forbindelse med at det ikke var mulig å opprette *circular arch members* med negativ radius i versjon 3.0 så ble det forsøkt å vise en alternativ metode for å tegne opp *arch members* som krummer nedover. Det virket ikke som den alternative metoden var direkte intuitiv, så utviklerteamet endte opp med å fortsatt tillate at *arch members* opprettes med negativ radius.



#### **-Flytting av arch members vs endring av høyde/radius for arch members:**



#### **-Har du hatt problemer med å zoome inn/ut?**

Ingen har hatt store problemer med dette.

#### **-Hva bør skje om man forsøker å kjøre analyse uten å ha definert lastkombinasjon?**

Alle bortsett fra én mener at brukeren bør varsles om at ingen lastkombinasjon (ingen laster) er lagt til, og få mulighet til å legge til egenlast og/eller nyttelast før analysen kjøres. Derfor har man tatt hensyn til dette ved å varsle brukeren med en dialogboks når ingen laster er lagt til. Dialogboksen gir mulighet for å legge til egenlast og kjøre analyse eller kjøre analysen uten laster.

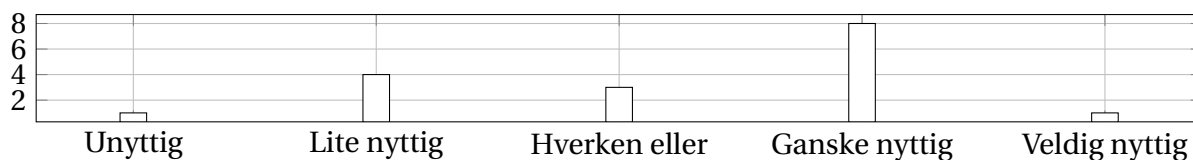
#### **-Er du klar over at man kan se kreftene og forskyvningene til et element ved å høyreklikke på det i resultatdiagrammet?**

Omtrent halvparten (8 stk) har blitt klar over det av seg selv, mens resten har funnet det ut vha. hjelp fra andre. Det var ikke noen som ikke var klar over det.

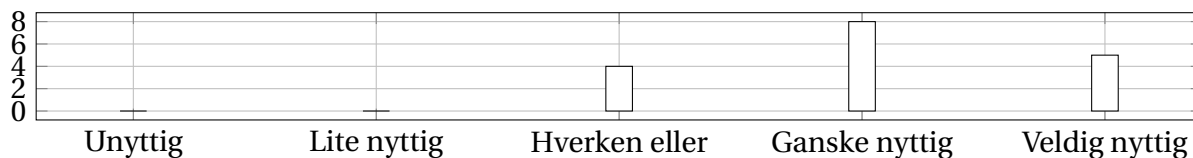
#### **-Synes du det er logisk å bruke høyre musetast til å velge elementer?**

13 studenter mener at det er logisk, mens de 4 siste heller ville brukt venstre musetast. Dette gjør at vi fortsetter med å bruke høyre musetast.

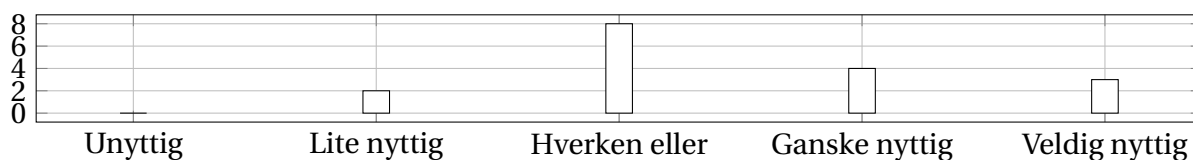
#### **-Hvilke nye funksjoner kunne du tenke deg?**



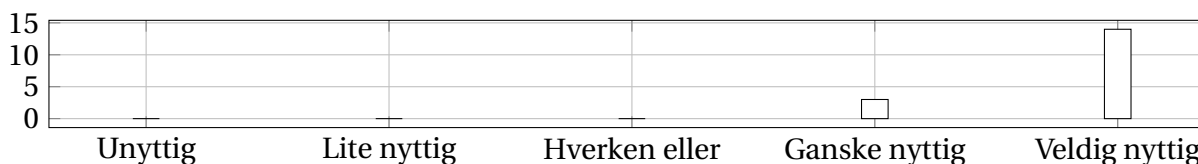
Figur B.1: Tutorial.



Figur B.2: Eksport av resultater til tekstfil.



Figur B.3: Import av modellgeometri (joints, members) fra andre programmer.



Figur B.4: Markering av maksimalverdier i resultatdiagrammer.

Testgruppen fikk også mulighet med å komme med andre forslag til ny funksjonalitet. Én student ønsket å få ut forskyvningsvektoren til hvert element, ikke bare  $x$ - og  $z$ -komponenten hver for seg. To stykker ønsket bedre visualisering av utnyttelsen av konstruksjonsdeler i hht. ståldimensjonering i EC 3-1-1 [2].

***-Fikk du løst øvingsoppgavene i KMEK ved å bruke fap2D?***

De aller fleste har fått løst øvingsoppgavene i faget, med forbehold om at de har måttet starte programmet på nytt opptil flere ganger. Noen studenter har ikke fått vist tekst i resultatdiagrammene slik at det har vært vanskelig å få lest ut maksimale krefter og forskyvninger, men det har ordnet seg ved at de har klikket på elementene og lett seg frem til de med maksimale verdier.

***-Hvilket program brukte du hovedsakelig i KMEK?***

Alle studentene brukte **fap2D**.

***-Dersom fap2D hadde vært mer feilfritt, hvilket program ville du foretrukket å bruke?***

Se forrige spørsmål.

## B.2.6 Konklusjon

Antallet henvendelser fra studenter var høyt t.o.m. februar, deretter avtok antallet. Dette var enten et tegn på at studentene gikk over til å bruke **Focus Konstruksjon** eller på at de ikke trengte hjelp lenger. Av de som ikke har svart på evalueringen er det sannsynligvis noen som gjorde det første, mens ellers er nok det siste sant.

Når forventningene sammenlignes med svarene fra evalueringen blir man overveldet over at alle som svarte har benyttet **fap2D** helt og holdent. Alle som svarte har fått gjennomført øvingsoppgavene, selv om de har hatt noen problemer.

Det hadde vært ønskelig å få inn flere evalueringsskjemaer, men siden evalueringen ble foretatt sent (etter forelesningsslutt i faget) må man tross alt være fornøyd. Utvalget er stort nok til at det kan trekkes noen slutninger fra svarene.

De generelle spørsmålene i evalueringen bekreftet at **fap2D** er et program som passer godt for målgruppen. De spesifikke spørsmålene var veldig nyttige når det skulle avgjøres om modifikasjoner skulle utføres. Derfor tolkes betatestingen som en suksess og neste versjon av **fap2D** kan lanseres som en ordinær utgivelse.

## B.2.7 Feilrapporteringskjema

### Feilrapportering for fap2D

---

Øving	<input type="text" value="1"/>
Oppgave	<input type="text"/>
Handling som forårsaket feilen:	<input type="text"/>
Feilmelding/programmets reaksjon:	<input type="text"/>
Hvilken fane i programmet var aktiv da feilen oppstod?	<input type="text" value="1 Modeling"/>
Legg ved et skjermbilde av feilen og/eller feilmeldingen	<input type="button" value="Velg fil"/> Ingen fil valgt
Legg til fap2D-filen, med feil	<input type="button" value="Velg fil"/> Ingen fil valgt
PC	<input type="text" value="Egen PC"/>
Operativsystem	<input type="checkbox"/> Windows 7 <input type="checkbox"/> Windows 8 <input type="checkbox"/> Eldre Windows <input type="checkbox"/> 32 bits <input type="checkbox"/> 64 bits
Email (frivillig):	<input type="text"/>

## B.2.8 Evalueringsskjema

### Evaluering av fap2D

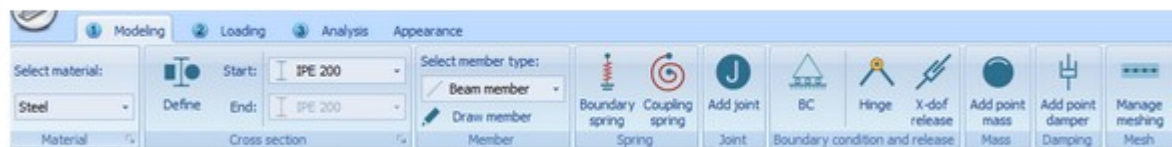
Hvilket operativsystem ble fap2D installert på? \*

- Windows 7
- Windows 8
- Annet OS
- 64-bits system
- 32-bits system

Hvordan gikk installasjonen av fap2D? \*

Fungerte installasjonsfilene greit? setup.exe, AccessDatabaseEngine, Microsoft .NET 4.5.

### Modelling



Har du brukt noen av følgende modelleringsobjekter i fap2D? \*

- Circular arch beam member
- Parabolic arch beam member
- Boundary spring
- Coupling spring
- Hinge
- X-dof release
- Point mass
- Point damper

Var det vanskelig å forstå hvordan noen av modellobjektene skulle brukes?

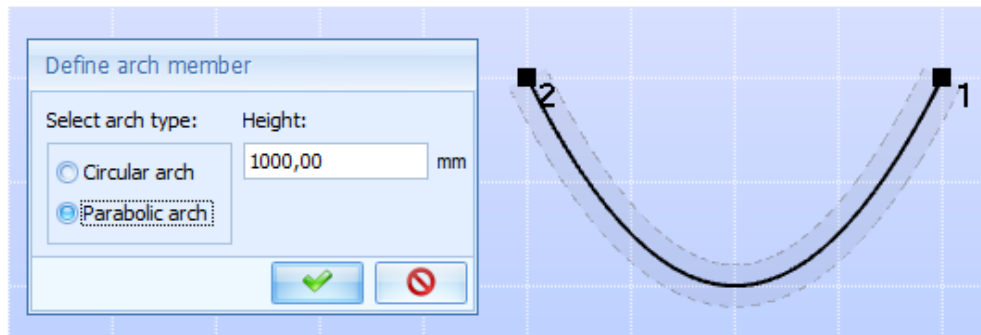
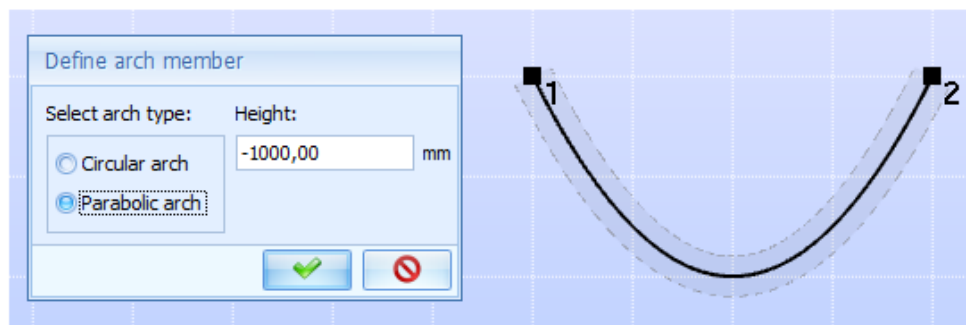
Var eventuelt informasjonen i tooltipsene til hjelp?



**Lage arch members som krummer nedover: \***

I nåværende versjon av fap2D er det problematisk å lage en circular arch member med negativ radius. For å omgå problemet vurderer vi å hindre brukeren fra å legge inn negativ radius for arch members. Det vil si at man må taste inn en positiv radius, og tegne arch members fra høyre mot venstre for å lage en arch member som krummer nedover. Hvor intuitiv vil du si at denne løsningen er?

- Lite intuitivt, enklere å taste inn negativ radius og tegne fra venstre mot høyre.
- Lite intuitivt, men dersom et tooltips/hjelpetekst kan beskrive prosedyren går det fint.
- Intuitivt.

**Tegne fra høyre mot venstre:****Tegne fra venstre mot høyre:****Flytting av arch members vs endring av høyde/radius for arch members \***

Hva synes du er mest hensiktsmessig?

- Å bruke musepekeren til å kunne flytte en arch member ved å markere og dra den. (Dette er mulig nå)
- Å bruke musepekeren til å kunne endre høyden/radius til en arch member ved å markere og dra den. (Dette vil si at dersom man ønsker å flytte en arch member må man flytte den ved å flytte member-ens joint i stedet).

Andre:

**Har du hatt problemer med å zoome inn/ut i fap2D?**

Hvis ja, spesifiser:

## Analysis

Hva bør skje dersom man forsøker å kjøre en analyse uten å ha definert en lastkombinasjon? \*

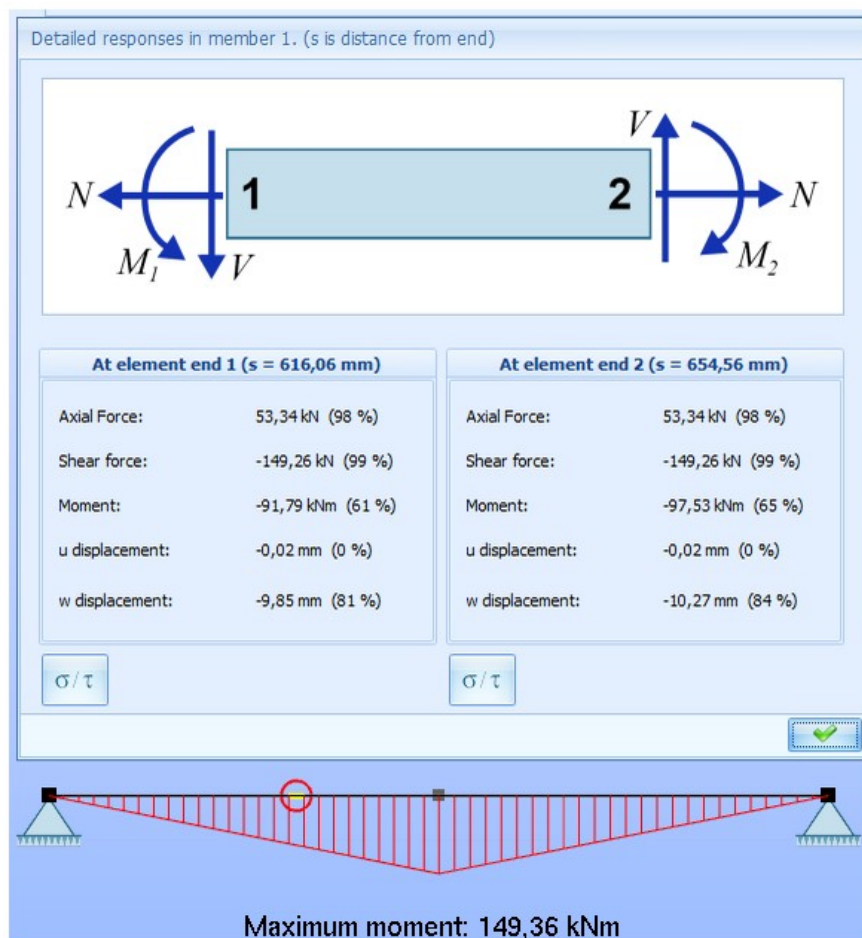
- Analysen kjøres med egenlast som default lastkombinasjon.
- Analysen kjøres uten noen laster. Resultatdiagrammene vil vises uten forskyvninger og krefter.
- Brukeren varsles om at ingen lastkombinasjon er lagt til og får mulighet til å legge til egenlast og/eller nyttelast før analysen kjøres.

## Results

Er du klar over at man kan se kreftene og forskyvningene til et element ved å høyreklikke på det i resultatdiagrammene? \*

Se figuren under.

- Ja, jeg fant det ut selv.
- Ja, jeg fant det ut vha hjelp fra User's manual eller tips fra andre.
- Nei



Synes du det er logisk å bruke høyre musetast til å velge elementer? \*

- Ja
- Nei, mer logisk å bruke venstre musetast
- Vet ikke.

## Ny funksjonalitet

Hvilke nye funksjoner kunne du tenke deg i fap2D? \*

	Unyttig	Lite nyttig	Hverken eller	Ganske nyttig	Veldig nyttig
Tutorial.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eksport av resultater til tekstfil (txt-fil, Word, Excel).	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Import av modellgeometri (joints, members) fra andre programmer.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Markering av maksimale resultatverdier (forskyvning, moment osv.) i resultatdiagrammene.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Andre funksjoner du kunne tenke deg?**

Funksjoner som er nyttige for nye brukere, for effektivitet, brukervennlighet osv.

## Generelt

**Fikk du løst øvingsoppgavene i KMEK Beregningsmetoder ved å bruke fap2D?**

Hvis ikke, nevnt kort hva som var utfordrende?

**Hvilket program brukte du hovedsakelig i KMEK Beregningsmetoder? \***

- fap2D
- Focus Konstruksjon

**Dersom fap2D hadde vært mer feilfritt (ingen feilmeldinger og andre irritasjonsmomenter), hvilket program ville du foretrukket å bruke? \***

- fap2D
- Focus Konstruksjon

# Tillegg C

## Dokumentasjon for utviklere

### C.1 Fargebruk i fap2D

#### Fargebruk i fap2D

For å sikre god lesbarhet og unngå forvirring i *GUI* er det viktig å gjøre bevisste fargevalg. Dokumentet viser en oversikt over farger som er i bruk i *modelling/results panel* i **fap2D**. Fargene endres i klassen `GlobalSettings`.

#### *Members, joints og diverse structural items*

- **Beams** Svart brukes fordi *beams* er mye brukt og bør ha nøytral farge.
- **Arch beams** *Arch beams* skiller seg ut pga. geometri, så svart brukes her óg.
- **Bar** Fargen symboliserer at *bars* kan ta både strekk- og trykkrefter.
- **Cable** Fargen symboliserer at *cables* bare tar strekkrefter.
- **Strut** Fargen symboliserer at *struts* bare tar trykkrefter.
- **Formfeil** Fargen brukes for *members* med lokal formfeil.
- **Load path** Fargen brukes når en *member* er del av en *load path*.
- **Joint** Svart brukes fordi *joints* er mye brukt og bør ha en nøytral farge. Geometrien skiller dem fra de svarte *members*.
- **Internal joint**
- **Markert joint**
- **Markert internal joint**
- **Overlappende joints** Symboliserer overlappende *joints*. Rød er passende som varselfarge.

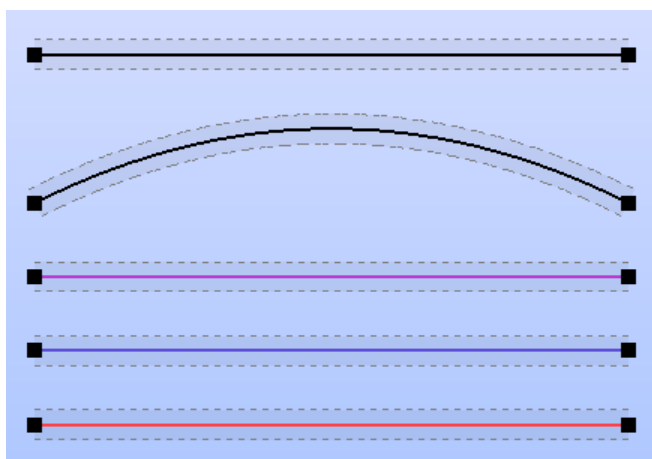
```

JointColor = new Color4f(0.0f, 0.0f, 0.0f, 1.0f);
InternalJointColor = new Color4f(0.392f, 0.392f, 0.392f, 1.0f);
BeamMemberColor = new Color4f(0.0f, 0.0f, 0.0f, 0.5f);
BarMemberColor = new Color4f(0.75f, 0.25f, 0.75f, 0.5f);
StrutMemberColor = new Color4f(0.16f, 0.16f, 1.0f, 0.5f);
CableMemberColor = new Color4f(1.0f, 0.16f, 0.16f, 0.5f);
MemberEdgeColor = new Color4f(0.5f, 0.5f, 0.5f, 0.05f);
CrossSectionColor = new Color4f(0.0f, 0.0f, 0.0f, 0.15f);
LocalImperfectionColor = new Color4f(1.0f, 0.8f, 0.0f, 0.05f);
LoadColor = new Color4f(0.0f, 0.7f, 0.0f, 0.2f);
ReleaseColor = new Color4f(0.16f, 0.42f, 0.55f, 0.0f);
SpringColor = new Color4f(1.0f, 0.0f, 0.0f, 0.0f);

```

Fargeverdier for joints og members i **fap2D**.

I **fap2D 3.0.0 Beta** var både *bars*, *cables* og *struts* farget lilla. I **fap2D 3.1** er dette endret. Endringen fører til at koden blir mindre effektiv (tre nye if-setninger), men det vil nok ikke ha mye å si for ytelsen. Selv om **rød** og **blå** kan forveksles med fargene som brukes for å vise strekk- og trykkrefter i resultatdiagrammene så er ikke det farlig, for disse fargene vil bare vises under modellering.



Fra toppen: Beam, arch beam, bar, strut, cable.



Member med lokal formfeil tegnes gul.



Member som har load path.

*Member outline* viser høyden på *member*-ens *cross section*. Den vises med en svart, men nesten transparent farge. Når *members* eller *submembers* markeres vil fargen skifte til lys gul. Dersom man legger flere *members* over hverandre vil fargen bli mørkere slik at brukeren kan se at det er flere.



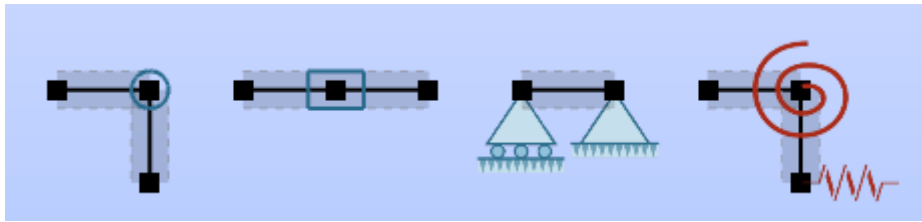
Markert member.



Fire members over hverandre fører til mørkere omriss.



Farger for joints: F.v.: Vanlig, overlappende, markert og internal (i midten av member-en).



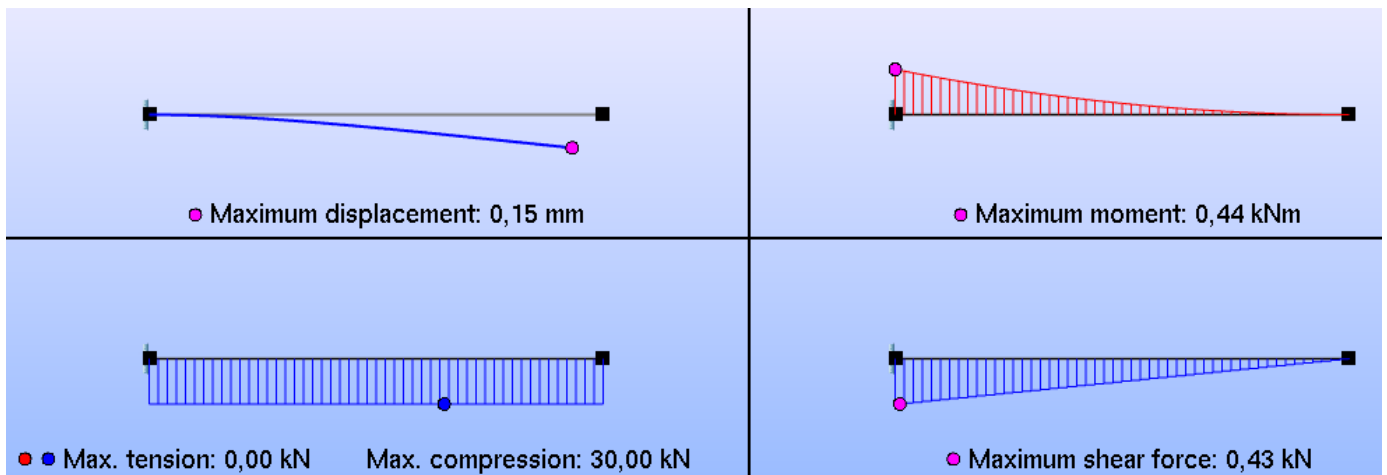
Diverse structural items.

## Resultater - krefter

- Forskyvde elementer (NegativeSectionForceColor)
- Markert element
- Strekkrefter (PositiveSectionForceColor)
- Trykkrefter (NegativeSectionForceColor)
- Momenter (PositiveSectionForceColor)
- Maksimalpunkter i diagrammer (NeutralForceColor)

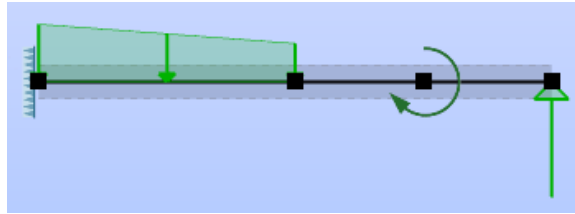
```
PositiveSectionForceColor = new Color4f(1.0f, 0.0f, 0.0f);
NegativeSectionForceColor = new Color4f(0.0f, 0.0f, 1.0f);
NeutralForceColor = new Color4f(1.0f, 0.0f, 1.0f);
ReactionForceColor = new Color4f(1.0f, 0.0f, 0.0f);
```

Fargeverdiene for resultater (krefter og forskyvning) i **fap2D**.



Farger i resultatdiagrammer.

## Loads



Loads tegnes grønne.

## Steel design usage

Utnyttelsen i hht. til ståldimensjonering vises vha. en fargeskala. Alle utnyttelse over 100 % tegnes rød. Ellers skifter skalaen farge for hver 10 %.



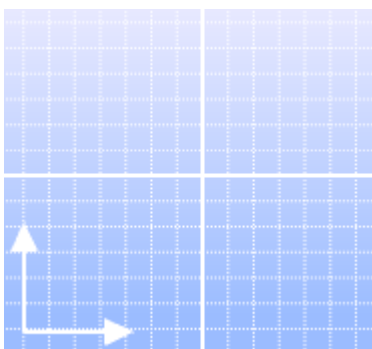
Fargeskalaen for steel design usage.

```
SteelDesignScaleColors = new Color4f[11];
SteelDesignScaleColors[0] = new Color4f(0.0f, 0.2f, 1.0f); // Blue
SteelDesignScaleColors[1] = new Color4f(0.0f, 0.71f, 1.0f); // Light blue
SteelDesignScaleColors[2] = new Color4f(0.0f, 1.0f, 0.71f); // Turquoise
SteelDesignScaleColors[3] = new Color4f(0.0f, 1.0f, 0.0f); // Green
SteelDesignScaleColors[4] = new Color4f(0.61f, 1.0f, 0.0f); // Light green
SteelDesignScaleColors[5] = new Color4f(0.82f, 1.0f, 0.0f); // Green yellow
SteelDesignScaleColors[6] = new Color4f(1.0f, 1.0f, 0.0f); // Yellow
SteelDesignScaleColors[7] = new Color4f(1.0f, 0.82f, 0.0f); // Dark yellow
SteelDesignScaleColors[8] = new Color4f(1.0f, 0.59f, 0.0f); // Orange
SteelDesignScaleColors[9] = new Color4f(1.0f, 0.31f, 0.0f); // Dark orange
SteelDesignScaleColors[10] = new Color4f(1.0f, 0.0f, 0.0f); // Red
```

Fargeverdier for steel design usage i **fap2D**.

## Grid, bakgrunn og aksekors

Brukeren kan endre fargen på *grid*, bakgrunn og aksekors. I utgangspunktet er aksekors og *grid* farget hvitt, mens bakgrunnen er en gradient med lyseblå nyanser.



Aksekors, grid og bakgrunn.

```
AxisColor = new Color4f(1.0f, 1.0f, 1.0f);
GridColor = new Color4f(1.0f, 1.0f, 1.0f);
BackgroundTopColor = new Color4f(0.92f, 0.92f, 1.0f);
BackgroundBottomColor = new Color4f(0.61f, 0.74f, 1.0f);
```

Fargeverdier for grid, aksekors og bakgrunn.

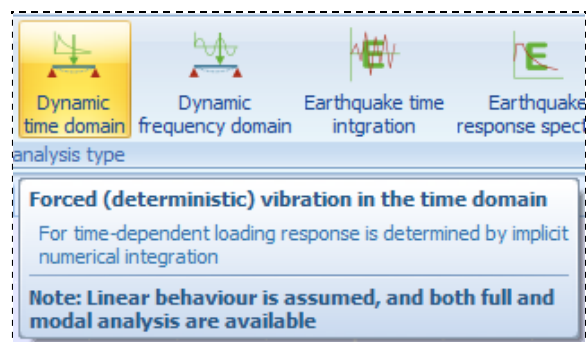
## C.2 Reviderte *tooltips*

# Reviderte *tooltips*

## Bakgrunn for revisjon

Mange funksjoner/knapper i **fap2D** benytter seg av *tooltips* for å forklare funksjonaliteten sin mer presist. En del *tooltips* er tilfredsstillende, men mange kan forbedres:

- Noen knapper bør vise mer informasjon i sitt *tooltip*.
- Mange *tooltips* er dårlig formaterte.
- Noen *tooltips* er dårlig formulerte.
- Noen *tooltips* stemmer dårlig overens med *User's manual*.
- Overskriften i noen *tooltips* er ikke samsvarende med navnet på knappen de tilhører. Overskriften bør alltid være tilnærmet lik navnet på knappen. Eksempel:



Det første punktet er en svakhet fordi ferske brukere ikke får den nødvendige informasjonen de trenger. De kan såklart lese gjennom *User's manual* for å lære seg programmet å kjenne, men det er nok ikke så mange som ønsker å lese gjennom en hel *User's manual* bare for å lære seg grunnleggende bruk.

De fire siste punktene er viktig for å gi brukerne inntrykk av at **fap2D** er profesjonelt og ryddig. Et ryddig og oversiktlig grensesnitt skaper ytterligere tillit til programmet, selv om det riktignok er analyseresultatene som er viktigst.

Samtidig som *tooltips* har blitt revidert har også en del knapper endret navn/formulering for å passe bedre til sitt *tooltip*. Det har også blitt fjernet overflødige ord, spesielt "Add", som egentlig ikke er nødvendige.

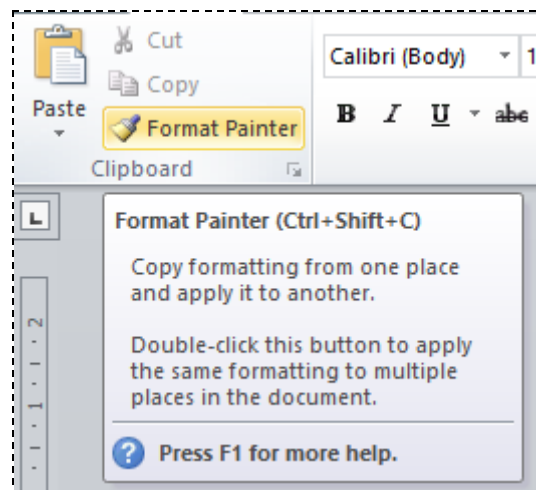
Det er blitt lagt til referanser til *User's manual* slik at brukeren enkelt kan finne ytterligere informasjon om de ulike knappene (spesielt analyseknappene).



### Formatering: *Super tooltips* i Microsoft Office Word

Figuren nedenfor viser et typisk *super tooltip* i **Word**, som vi bør etterligne for å få mest mulig ut av *tooltip*-funksjonaliteten i **fap2D**:

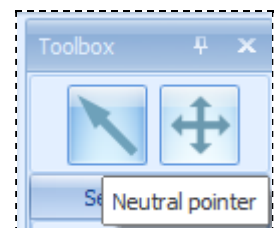
- Navn på knapp/funksjonalitet øverst, sammen med evt. hurtigtast.
- Generell informasjon i midten. Evt. forklaring vha. figur. Mest tekst her.
- Henvisning til relevante avsnitter i *User's manual* nederst, og spesielle merknader. Ikke mye tekst her.



Grunnen til at det bør være **lite tekst øverst og nederst** er at den uthevede skriften i disse to feltene vil ta all oppmerksomhet, når det faktisk er teksten i midten som er viktigst i *super tooltips*. Det er informasjonen i midten som skal sørge for at brukeren forstå hensikten med knappen/funksjonaliteten.

### Formatering: Vanlige *tooltips*

Vanlige *tooltips* er bare en sammenhengende streng med tekst. Den skal ikke være uthevet og skal gi akkurat nok informasjon til at brukeren forstår hensikten med en knapp/funksjonalitet. Dersom knappen bare er et symbol er det lurt å angi navnet på symbolet i *tooltip*-et. Om navnet ikke er nok til å forklare knappen/menyen kan det legges til en kort forklarende tekst.



I **C#.NET** kan man bruke egenskapen *TooltipTitle* for å angi navnet på en knapp, men vær klar over at *TooltipTitle* som regel ikke er synlig med mindre man også angir *Tooltip*.

## Keyboard shortcuts i fap2D

Alle *keyboard shortcuts*/hurtigtaster som brukes i programmet finnes i *User's Manual*. Hurtigtastene skal vises i *tooltip* for knapper som har samme funksjon som hurtigtasten.

### Keyboard shortcuts:

Press	Function
CTRL+T & CTRL+N	Make a new model
CTRL+S	Save the current model
CTRL+O	Open an existing model
CTRL+P	Print a picture of the current model
CTRL+D & Del	Delete marked objects in the model
CTRL+Z	Undo
CTRL+Y	Redo
CTRL+F4	Close the model
CTRL+TAB	Toggle between open models
CTRL+A	Mark all objects in the model
ESC	Close a dialog box / Reset mouse pointer
ENTER	Push the OK-button in dialog boxes
F1	Launch User's Manual

## Endringer fra fap2D 3.0.0 Beta til fap2D 3.1

Endringene som er gjort mellom disse versjonene er registrert vha. skjermbilder. Dersom noen av de nye *tooltips* skal revideres i framtiden kan det være nyttig å bruke dette dokumentet som sammenligning/referanse.

Endringer er gjennomført i følgende deler av **fap2D**:

- Statuslinjen, *status bar*, nederst i programmet.
- *Application menu* og *Quick access toolbar*
- *Toolbox panel*
- *Modeling ribbon page*
- *Loading ribbon page*
- *Analysis ribbon page*
- *Results ribbon page*
- *Settings*
- Diverse dialogbokser, men langt fra alle.

Vi har ikke prioritert å gå systematisk gjennom informasjonen i alle dialogboksene i programmet. Det ville tatt lang tid å sette seg inn i alle analysene slik at vi kunne kvalitetssikre informasjonen for hvert enkelt *tooltip*. Vi stoler på at de ansvarlige for implementasjon av hver enkelt analyse forklarer bruken av analysen i tilstrekkelig grad.

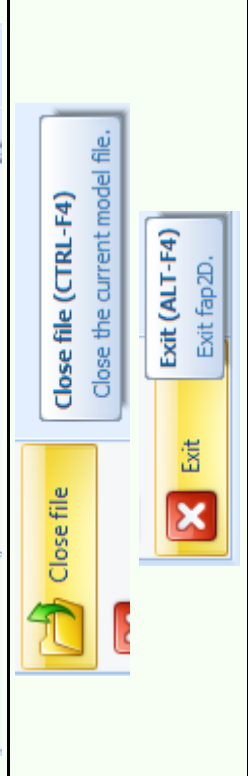
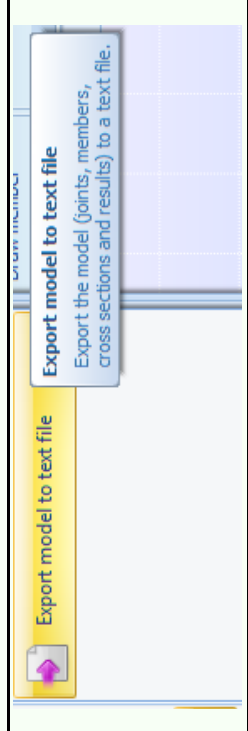
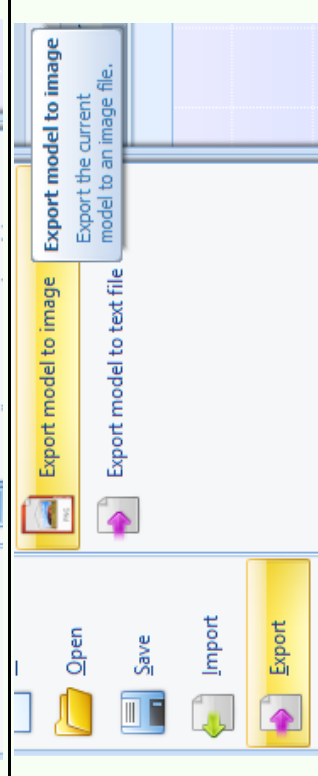
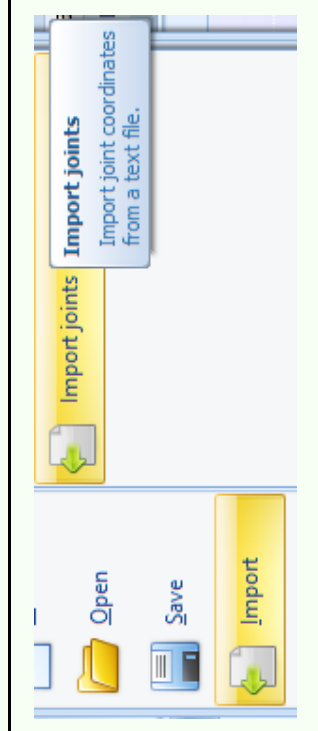
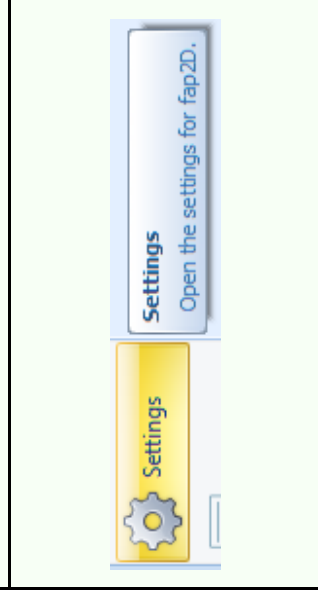
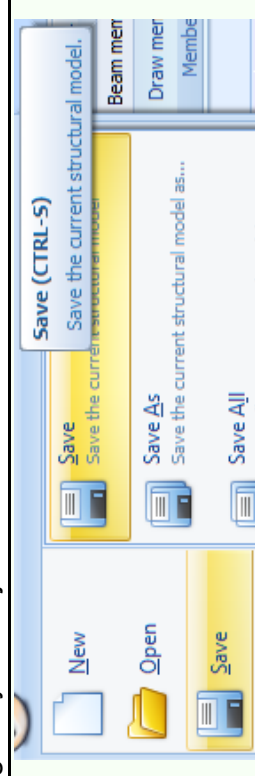
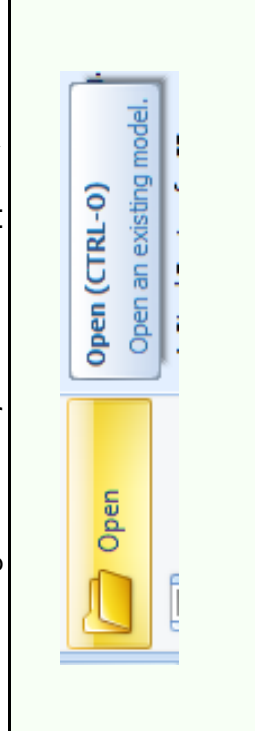
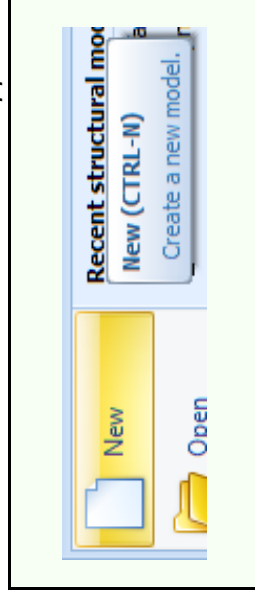
## Status bar, nederst i programmet

Her er det en knapp som brukes til printing og et tekstfelt som har vært brukt til å vise koordinatene til musepekeren i tidligere versjoner enn **fap2D 3.0**. Koordinatene er innført igjen i **fap2D 3.1**, med et *tooltip*. *Tooltip*-et gjentar bare innholdet i tekstfeltet, vi har ikke lagt til mer informasjon fordi det trengs ikke mer forklaring av dette feltet.



## Application menu

I *Application menu* manglet det *tooltips* for alle knappene, så det er lagt til nye i versjon **3.1**.



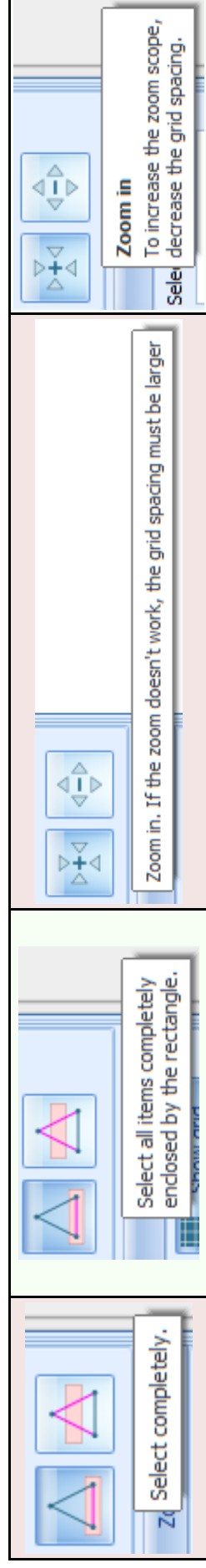
## Quick access toolbar


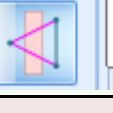
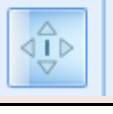

I Quick access toolbar er i orden med unntak av at det må legges til informasjon om hurtigtaster for printing, undo og redo.



## Toolbox panel




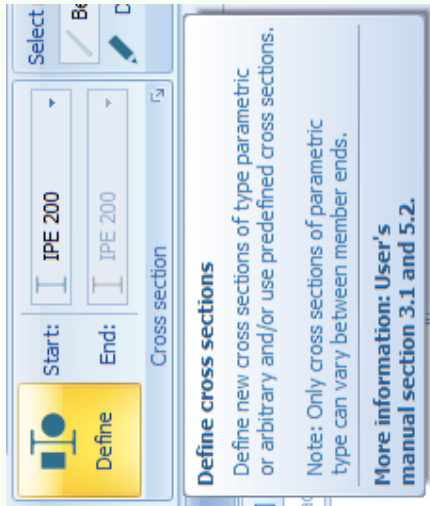

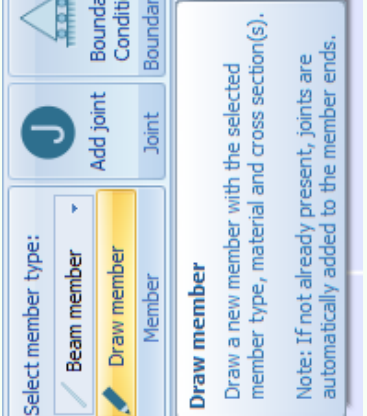
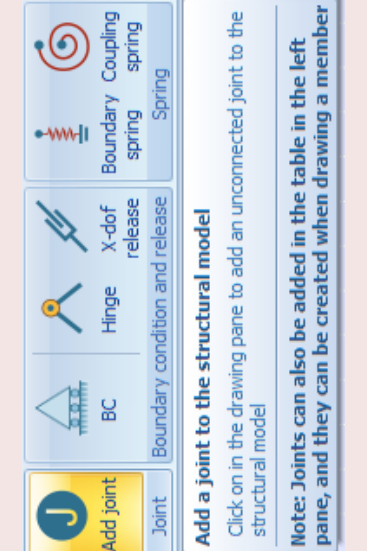
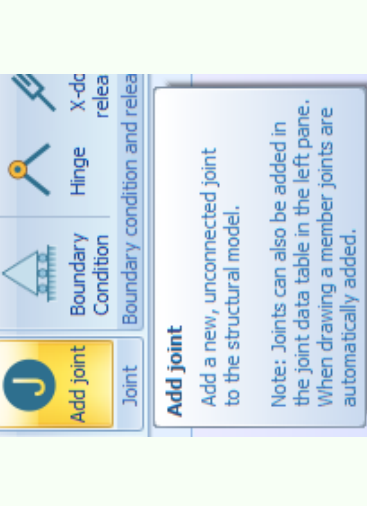
Under Selection og Zooming er det fire knapper for å markere *structural items* i modellen, som kan forklares mer presist. For *Member outline*-knappen i gruppen *ID and outline* vil vi være mer presise fordi *member outline* indikerer hvilke *cross sections* som er i bruk. Vi foreslo å endre navnet på knappen til noe å la *Show cross sections*. I samråd med veileder ble vi derimot enige om å ikke bruke en slik betegnelse fordi knappen indikerer bare de ekstreme fibrene til en *member*, ikke nøyaktig formen og egenskapene til *member*-ens *cross section*.



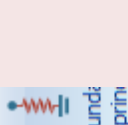
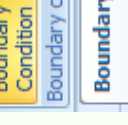


 <p><b>Select partial.</b></p>	 <p>Select all items enclosed by rectangle, partially or completely.</p>	 <p>Zoom out. If the zoom doesn't work, the grid spacing must be larger</p>	 <p><b>Zoom out</b> To increase the zoom scope, increase the grid spacing.</p>
<p><b>Member outline</b> Show extreme member fibres.</p>	<p><b>Member outline</b> Show extreme member fibres.</p>		



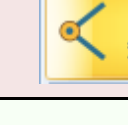
### Modeling ribbon page

Tooltips i modeling ribbon page har fått mer informasjon, bedre formatering og henvisninger til User's manual.


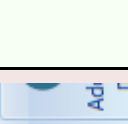

 <p>Select material: Steel Material</p> <p><b>Select or add material</b> Predefined materials cannot be edited <b>Note: Selected (current) material will be assigned to all new members</b></p>	 <p>Select material: Steel Material</p> <p><b>Select or add material</b> Select or define a new material. The selected material will be assigned to all new members. <b>Note:</b> Predefined materials cannot be edited. <b>More information:</b> User's manual section 5.2.</p>	 <p>Start: IPE 200 End: IPE 200 Cross section</p> <p><b>Define cross sections</b> Define new cross sections of type parametric or arbitrary and/or use predefined cross sections. <b>Note:</b> Only cross sections of parametric type can vary between member ends. <b>More information:</b> User's manual section 3.1 and 5.2.</p>	 <p>Start: IPE 200 End: IPE 200 Cross section</p> <p><b>Define cross sections</b> Define new cross sections of type parametric or arbitrary and/or use predefined cross sections. <b>Note:</b> Only cross sections of parametric type can vary between member ends. <b>More information:</b> User's manual section 3.1 and 5.2.</p>
 <p>Select member type: Beam member <b>Draw member</b> Member</p> <p><b>Draw member</b> Draw a new member with the selected member type, material and cross section(s). <b>Note:</b> If not already present, joints are automatically added to the member ends.</p>	 <p>Select member type: Beam member <b>Draw member</b> Member</p> <p><b>Draw member</b> Draw a new member with the selected member type, material and cross section(s). <b>Note:</b> If not already present, joints are automatically added to the member ends.</p>	 <p>Add joint Joint</p> <p><b>Add joint to the structural model</b> Click on in the drawing pane to add an unconnected joint to the structural model <b>Note: Joints can also be added in the table in the left pane, and they can be created when drawing a member</b></p>	 <p>Add joint Joint</p> <p><b>Add joint</b> Add a new, unconnected joint to the structural model. <b>Note:</b> Joints can also be added in the joint data table in the left pane. When drawing a member joints are automatically added.</p>

 Boundary condition and release  
 Hinge  
 X-dof release  
 Boundary spring

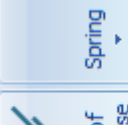
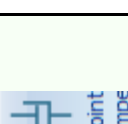
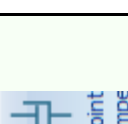
**Pick and apply BC and symbol**  
to one or more joints/internal joints.

 Boundary Condition  
 Hinge  
 X-dof release


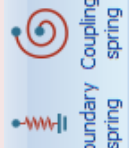

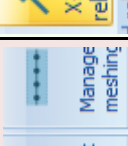
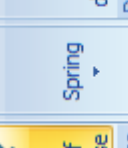
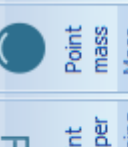
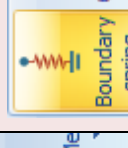
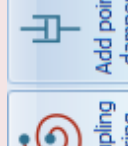
**Boundary condition**  
Add a boundary condition to a joint (or internal joint).  
 Note: Boundary conditions can be removed through a joint's right-click dialog.  
**More information:**  
 User's manual section 3.3.

 Hinge  
 X-dof release  
 Boundary spring  
 Coupling spring  
 Add point damper

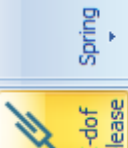

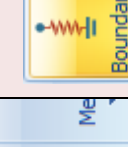

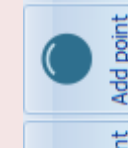

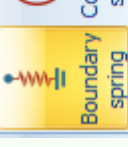
**Introduce rotational release(s) at one or more joints/internal joints.**  
 Uncouple rotation between member ends.  
**Note: Releases can be removed from the joint's right-click dialog.**

 Hinge  
 X-dof release  
 Spring  
 Point damper

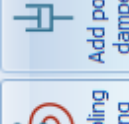
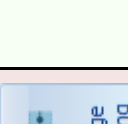
**Hinge**  
 Introduce rotational release(s) at a joint (or internal joint).  
 Uncouple rotation between member ends.  
 Note: Releases can be removed through a joint's right-click dialog.

 X-dof release  
 Boundary spring  
 Coupling spring  
 Add point damper  
 Add point mass  
 Point damper  
 Point mass  
 Manage meshing

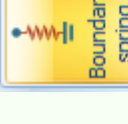
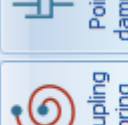
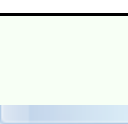
**Introduce a release of x-displacement at one or more joints/internal joints.**  
 Uncouple the displacement in x-direction (global or local) between member ends.  
**Note: Releases can be removed from the joint's right-click dialog.**

 X-dof release  
 Spring  
 Point damper  
 Coupling spring  
 Add point damper  
 Add point mass  
 Manage meshing




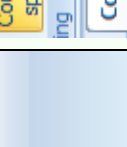
**X-dof release**  
 Uncouple displacements in current x-direction (local if transformed, otherwise global) between member ends.  
 Note: Releases can be removed through a joint's right-click dialog.

 Boundary spring  
 Coupling spring  
 Add point damper  
 Add point mass  
 Manage meshing



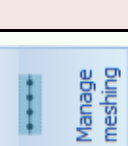

**Boundary spring**  
 Define stiffness of an elastic spring and apply it to the appropriate DOF at any joint.  
**Note: The spring is fixed at the other end.**

 Boundary spring  
 Coupling spring  
 Point damper

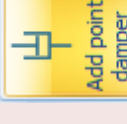

**Boundary spring**  
 Define the stiffness of an elastic spring and apply it to an appropriate DOF at a joint.  
 Note: The spring is fixed at the other end.

 Coupling spring  
 Add point damper  
 Add point mass  
 Manage meshing




**Coupling spring**  
 Define stiffness of a elastic spring and apply it between two member ends at joint with appropriate displacement release(s)

 Coupling spring  
 Point damper  
 Point mass  
 Manage meshing

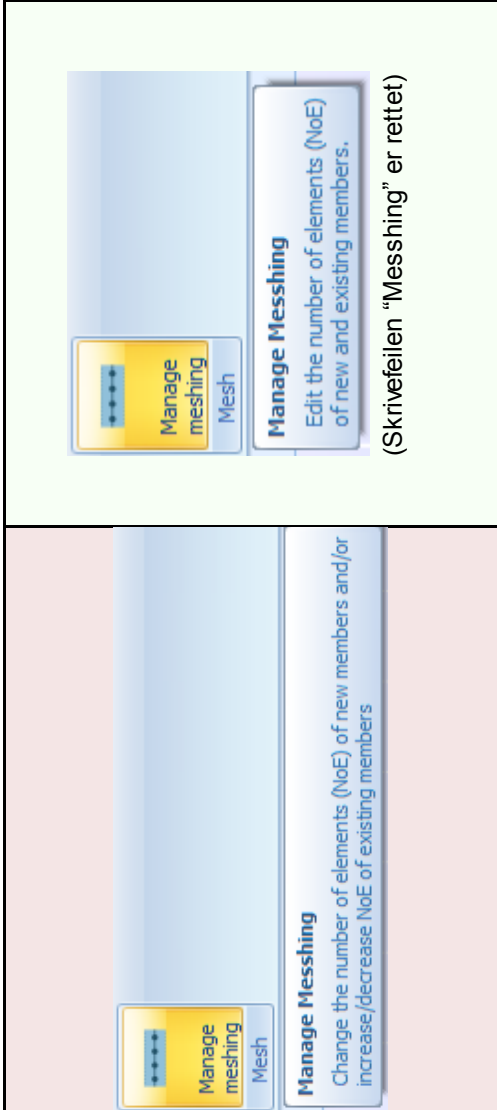
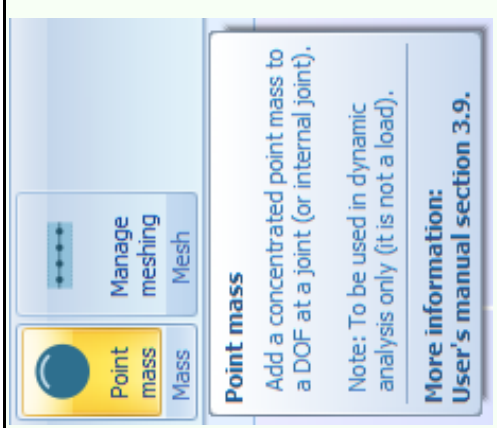
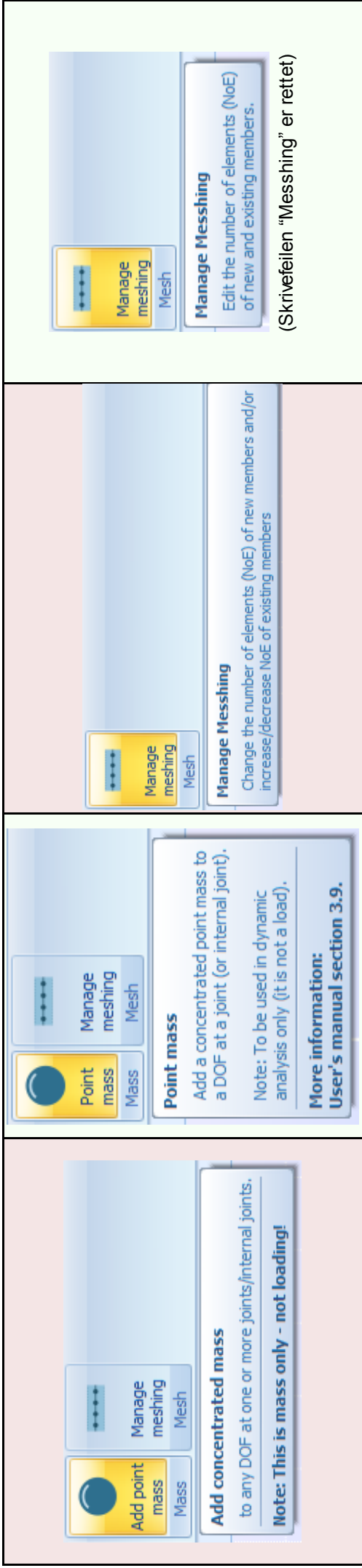
**Coupling spring**  
 Define the stiffness of an elastic spring and apply it between two member ends at a joint with displacement release(s).  
 Note: To be used in combination with Hinge and/or X-dof release.  
**More information:**  
 User's manual section 3.3 and 4.3.

 Add point damper  
 Add point mass  
 Manage meshing

**Add concentrated damping**  
 to any DOF at one or more joints/internal joints.

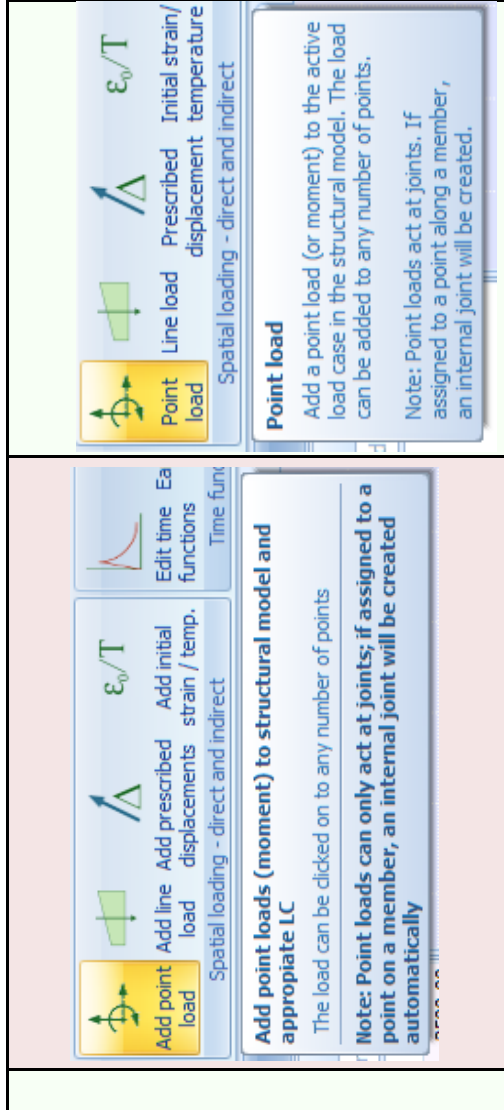
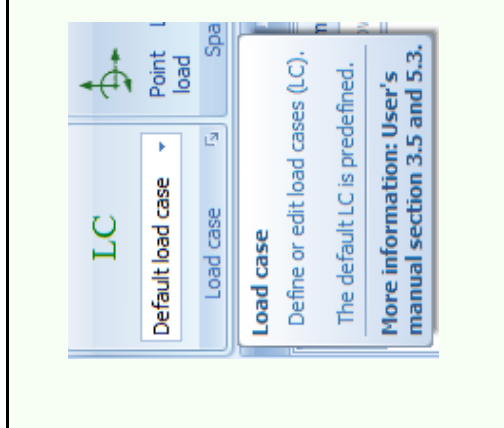
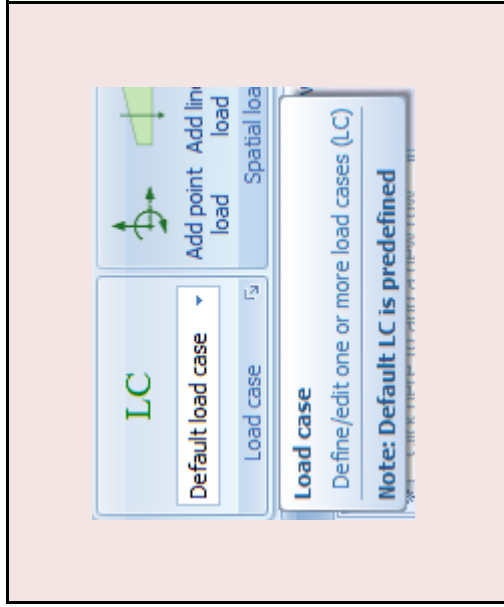
 Point damper  
 Point mass  
 Manage meshing

**Point damper**  
 Add a concentrated point damper to a DOF at a joint (or internal joint).



*Loading ribbon page*

Tooltips i loading ribbon page har fått mer informasjon, bedre formatering og henvisninger til User's manual.



 Add line load  
 initial loading - direct and indirect


 Add prescribed displacements - direct and indirect


 Add initial strain / temp.

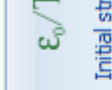
 Edit time functions  
 Time functions


 Earthquakes


**Add a line load of appropriate type to one or more members/sub-members and appropriate LC**  
 Click: specified load on to member(s)  
**Note: Arrow in load symbol specifies load direction**

 Line load  
 initial loading - direct and indirect


 Prescribed displacement - direct and indirect


 Add initial strain/temperature


 Edit time functions  
 Time functions


 Earthquakes

**Line load**  
 Add a line load to the active load case in the structural model.  
 Apply the load by clicking it on to the chosen members and/or submembers.


 Add prescribed displacements - direct and indirect


 Add initial strain / temp.


 Edit time functions  
 Time functions


 Earthquakes

**Prescribed displacement**  
 Prescribed displacements can be specified for fixed dofs only.  
**Note: An LC named "Prescribed displ." is automatically created when a prescribed displacement is specified - this LC can be included in any LCmb. All prescribed displacements are associated with this LC**

 Prescribed displacement - direct and indirect

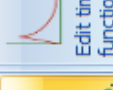
 Add initial strain/temperature


 Edit time functions  
 Time functions

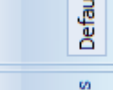
 Earthquakes

**Prescribed displacement**  
 Add a prescribed displacement to a fixed DOF.  
 Note: All prescribed displacements will be included in a load case called "Prescribed displ."  
**More information: User's manual section 3.5 and 5.3.**


 Add initial strain / temp.


 Edit time functions  
 Time functions


 Earthquakes


 LCmb  
 Default load combination  
 Load combination

**Add initial strain (temperature) to one or more members/sub-members**  
 Click the specified loading on to member(s)  
**Note: All loading of this type is assigned to the same (predefined) LC named "Initial strain"; this LC can be included in any LCmb**

 Add initial strain/temperature - indirect

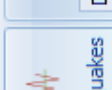
 Edit time functions  
 Time functions

 Earthquakes

 LCmb  
 Default load combination  
 Load combination


**Initial strain (e.g. temperature)**  
 Add initial strain to a member/submember, by clicking the specified load on to member(s).  
 Note: All initial strain loading will be included in a load case called "Init. strain".  
**More information: User's manual section 3.5 and 5.3.**


 Edit time functions  
 Time functions

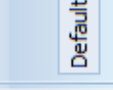
 Earthquakes

 LCmb  
 Default load combination  
 Load combination


**Define one or more "time" functions**  
 Time functions are relevant for nonlinear static or time domain dynamic analysis

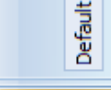
 Edit time functions  
 Time functions

 Earthquakes

 LCmb  
 Default load combination  
 Load combination

**Time functions**  
 Define or edit time functions to be used in dynamic time domain analysis.  
 Note: It is currently not possible to use custom time functions in nonlinear static analysis, only predefined time functions.  
**More information: User's manual section 3.6.**

 Earthquakes  
 functions

 LCmb  
 Default load combination  
 Load combination


Show all loads  
 Show selected LC  
 Show selected LC  
 View Loading

**Choose predefined or import time-series of earthquake ground accelerations**  
 9 predefined earthquakes are available

 Earthquakes  
 functions


 LCmb  
 Default load combination  
 Load combination

**Earthquakes**  
 Choose a predefined ground acceleration time series or import a time series from file.  
 Each time series represents one component/direction.  
**More information: User's manual section 3.6 and 6.9.**

 LCmb  
 Default load combination  
 Load combination

Show all loads  
 Show selected LC  
 Show selected LCmb  
 View Loading

**Load combination**  
 Define/edit one or more load combinations (LCmb)  
**Note: Default LCmb is predefined**

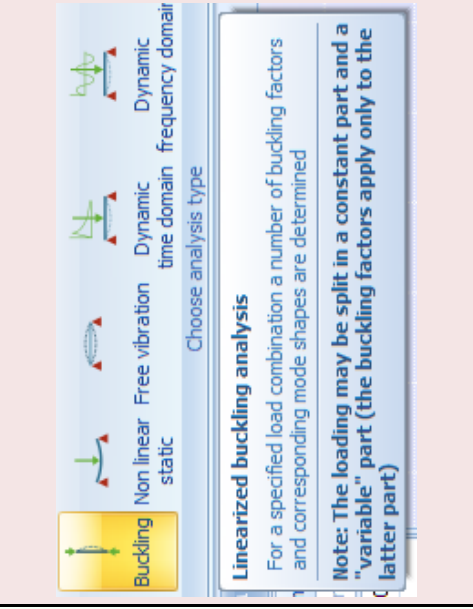
 LCmb  
 Default load combination  
 Load combination

**Load combination**  
 Define or edit load combinations (LCmb).  
 Note: The default load combination is predefined.  
**More information: User's manual section 3.5 and 5.3.**



## Analysis ribbon page

Tooltips in analysis ribbon page har fått mer informasjon, bedre formatering og henvisninger til *User's manual*.

 <p><b>Linear static analysis</b> for one load combination</p> <p>Note: Bi-linear axial members (cables and struts) are admissible in the model</p> <p>More information: User's manual section 6.1.</p>	 <p><b>Linear static</b></p> <p>Linear static analysis for one load combination.</p> <p>Note: Bi-linear axial members (cables and struts) are permitted in the model.</p> <p>More information: User's manual section 6.1.</p>	 <p><b>Compute influence lines for specified response parameters</b></p> <p>Note: The model must be completely linear. Bi-linear axial members (cables and struts) are not permitted.</p> <p>More information: User's manual section 6.2.</p>	 <p><b>Influence lines</b></p> <p>Compute influence lines for the specified response parameters.</p> <p>Note: The model must be completely linear. Bi-linear axial members (cables and struts) are not permitted.</p> <p>More information: User's manual section 6.2.</p>
 <p><b>"Real time" analysis of a completely linear model subjected to a "load train" moving over members comprising a "load path".</b></p> <p>Response diagrams change in step with the train movement.</p> <p>More information: User's manual section 6.3.</p>	 <p><b>Real time analysis</b></p> <p>"Real time" analysis of a completely linear model subjected to a <i>load train</i> moving over members comprising a <i>load path</i>.</p> <p>Response diagrams change in step with the <i>load train's</i> movement.</p> <p>More information: User's manual section 6.3.</p>	 <p><b>Linearized buckling analysis</b></p> <p>For a specified load combination a number of buckling factors and corresponding mode shapes are determined</p> <p>Note: The loading may be split into a constant part and a "variable" part (the buckling factors apply only to the latter part)</p>	 <p><b>Linearized buckling</b></p> <p>Compute a number of buckling factors and corresponding mode shapes for a specified load combination.</p> <p>Note: The loading may be split into a constant and a variable part. The buckling factors apply to the variable parts only.</p> <p>More information: User's manual section 6.4.</p>



Shear deformation

**LCmb**

Default load combination

Load combination

Run Analysis

Linear static anal

**Include shear deformations**

In order to include shear deformations in the analysis this button needs to be pressed down

**Note: By default shear deformations are neglected**

Shear deformation

**LCmb**

Default load combination

Load combination

Run Analysis

Linear static anal

**Include shear deformations**

To include shear deformations in the analysis this button has to be pressed down.

Note: By default shear deformations are neglected.

Influence line

Define load path

Define load train

Run analysis

Influence line analysis

**Define a "load path" and a "travelling load" as basis for influence lines for response parameters**

A load path consists of one or more (connected) members and the travelling ("influence") load acts in the direction of one of the global axes

Influence line

Define load path

Define load train

Run analysis

Influence line analysis

**Define load path**

Define a *load path* and a *travelling load* as basis for influence lines for response parameters.

A *load path* consists of one or more (connected) members and the *travelling (influence) load* acts in the direction of one of the global axes.

Influence line analysis

Run analysis

Define load train

Define load path

**Define one or more "load trains" to travel the influence load path**

A load train consists of an arbitrary number of concentrated forces each acting at a given distance from the first and all forces act in a direction parallel with the "influence load"

Influence line analysis

Run analysis

Define load train

Define load path

**Define load train**

Define one or more *load trains* to travel the *influence load path*.

A *load train* consists of an arbitrary number of concentrated forces each acting at a given distance from the first force. All forces act in a direction parallel with the *influence load*.

Real time

Define load path

Define load train

Run analysis

Real time analysis

**Define a "load path" for real time analysis of a "load train" (B) travelling the path**

This load path is more general (less restrictive) than the "influence line path"

Real time

Define load path

Define load train

Run analysis

Real time analysis

**Define load path**

Define a *load path* for "real time" analysis of a *load train*, B, travelling the path.

This *load path* is more general (less restrictive) than the *influence line path*.

Define load train

Define one or more *load trains* to travel the real time *load path*.

This load train is more general than the *influence load train* (no restrictions and even concentrated moments can be included).

Number of load train positions: 100

Load train positions

Response

Displ.  $\delta$

Bending moment  $(M)$

All  $\delta$   $M$   $N$   $V$

**Define one or more "load trains" to travel the real time load path**

This load train is more general than the "influence load train" (no restrictions and even concentrated moments can be included)

Define load train

Define one or more *load trains* to travel the real time *load path*.

This load train is more general than the *influence load train* (no restrictions and even concentrated moments can be included).

Number of load train positions: 100

Load train positions

Response

Displ.  $\delta$

Bending moment  $(M)$

All  $\delta$   $M$   $N$   $V$

**Number of equally spaced positions of the load train, on the entire load path, for which results are computed**

Define load train

Define one or more *load trains* to travel the real time *load path*.

This load train is more general than the *influence load train* (no restrictions and even concentrated moments can be included).

Number of load train positions: 100

Load train positions

Response

Displ.  $\delta$

Bending moment  $(M)$

All  $\delta$   $M$   $N$   $V$

**Number of load train positions**

Number of equally spaced positions of the *load train*, on the entire *load path*, for which results are computed.

Define load train

Define one or more *load trains* to travel the real time *load path*.

This load train is more general than the *influence load train* (no restrictions and even concentrated moments can be included).

Number of load train positions: 100

Load train positions

Response


Displ.  $\delta$

Bending moment  $(M)$


All  $\delta$   $M$   $N$   $V$

**Number of load train positions**

Number of equally spaced positions of the *load train*, on the entire *load path*, for which results are computed.


 Loading  
 Loading

Number of modes:   
 Buckling modes

 Run Analysis  
 Buckling analysis

**Specify load combination for "variable" loading(subject to buckling factor and (optionally) a load combination representing a constant (dead) load (excluded from the buckling factor**


**Note: Must be specified - no defaults available**

 Loading  
 Loading

Number of modes:   
 Buckling modes


**Loading**  
 Specify load combination for *variable* loading, subjected to a buckling factor.  
 Optionally, specify a load combination representing a constant (dead) load, excluded from the buckling factor.  
 Note: Must be specified, no defaults are available.

Number of modes:   
 Buckling modes

 Run Analysis  
 Buckling analysis

**The number of (the lowest) buckling modes to be determined**

Number of modes:   
 Buckling modes

 Run Analysis  
 Buckling analysis


**Number of modes**  
 The number of (the lowest) buckling modes to be determined.


 Load history  
 Load History


 Shape imperfection  
 Geometry

 Run Analysis  
 Non linear static analysis


**Specify how the loading shall be applied**  
**Note: Must be specified - no defaults available**


 Load history  
 Load History

 Shape imperfection  
 Geometry

 Run Analysis  
 Non linear static analysis


**Load history**  
 Specify the load combinations to use in the nonlinear analysis.  
 Note: Must be specified, no defaults are available.

 Shape imperfection  
 Geometry

 Run Analysis  
 Non linear static analysis

**Specify geometric imperfection - shape and size - to include in the model**

 Shape imperfection  
 Geometry

 Run analysis  
 Run analysis

**Shape imperfection**  
 Specify geometric imperfection in terms of shape and size, to include in the model.


 Stiffness  
 Stiffness and mass

 Mass  
 Mass

Number of modes:   
 Shift value:   
 Frequencies and shift values

 Run Analysis  
 Free vibration


**Include geometric stiffness**  
 Default stiffness is material stiffness only. Geometric stiffness due to a specified LCmb may be included

 Stiffness  
 Stiffness and mass

 Mass  
 Mass


Number of modes:   
 Shift value:   
 Frequencies and shift values

**stiffness**  
 Use only material stiffness in the stiffness matrix, or include geometric stiffness as well.  
 The geometric stiffness will correspond to a specified applied loading.

 Mass  
 and mass

Number of modes:   
 Shift value:   
 Frequencies and shift values

**Select one of three mass models**  
 Lumped mass is default

 Mass  
 and mass

Number of modes:   
 Shift value:   
 Frequencies and shift values





**Mass**  
 Select one of three mass models. Lumped mass is default.  
**More information:**  
 User's manual section 3.9.

 Damping  
 Loading  
 Dyn. LCmb  
  $\Delta t$  Time domain



**Specify the damping**  
 Standard Rayleigh damping based on a damping ratio of 0,02 in the two lowest modes is default

 Damping  
 Loading  
 Dyn. LCmb

**Damping**  
 Choose damping model. Standard Rayleigh damping based on a damping ratio of 0,02 in the two lowest modes is default.  
**More information:**  
**User's manual section 3.10.**

 Computational model  
 Stiffness  
 Mass  
 Damping  
 Loading

**Computational model**  
 Choose between full (default) or modal analysis.

 Computational model  
 Stiffness  
 Mass

**Computational model**  
 Choose between full (default) or modal analysis.

Number of modes: 5  
 Shift value:  
 Run Analysis  
 Free vibration analysis

**Number of modes to be determined**  
 The modes are those that are (numerically) closest to the origin

Number of modes: 5  
 Shift value:  
 Run analysis  
 Run analysis



**Number of modes**  
 Number of modes and corresponding eigenfrequencies to be determined.  
 The modes are those that are (numerically) closest to the origin.

Number of modes: 5  
 Shift value:  
 Run Analysis  
 Free vibration analysis

**Shift value in Hz**  
 The shift value shifts the origin of the characteristic polynomial (up or down the frequency axis) enabling detection of modes close to the shifted origin

Number of modes: 5  
 Shift value:  
 Run analysis  
 Run analysis



**Shift value**  
 The shift value (in Hz) shifts the origin of the characteristic polynomial (up or down the frequency axis) enabling detection of modes close to the shifted origin.

 Loading  
 Dyn. LCmb  
  $\Delta t$  Time domain

**Specify dynamic load combination(s)**  
 Spatial load cases are combined with time functions  
**Note: Must be specified - no defaults available**

 Loading  
 Dyn. LCmb

**Loading**  
 Specify dynamic load combination(s); Spatial load cases are combined with time functions.  
 Note: Must be specified, no defaults are available.

 Dyn. LCmb  
  $\Delta t$  Time domain

**Choose dynamic LCmb to be analyzed**

 Dyn. LCmb



**Dynamic load combination**  
 Choose a dynamic load combination to be analysed.





Time domain  
 Time domain  
 Dynamic analysis time domain

**Specify key parameters for the time axis**  
**Note: Must be specified - no defaults available**

Loading  
 Dynamic analysis frequency domain

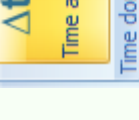
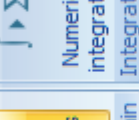
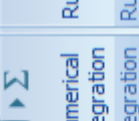
**Specify the loading which is harmonic (one or more frequencies) or periodic but non-harmonic**  
**Note: Must be specified - no defaults available**





Time domain  
 Numerical integration  
 Earthquake time integration analysis

**Specify key parameters for the time axis**  
**Note: Default values, relevant for the earthquake record, are available**

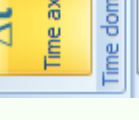
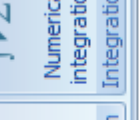

Time domain  
 Numerical integration  
 Run analysis

**Time axis**  
 Specify key parameters for the time axis.  
 Note: Must be specified, no defaults are available.





Loading  
 Run analysis

**Harmonic or periodic loading**  
 Specify the loading which is harmonic (one or more frequencies) or periodic, but non-harmonic.  
 Note: Must be specified, no defaults are available.

Time domain  
 Numerical integration  
 Run

**Time axis**  
 Specify key parameters for the time axis.  
 Note: Default values, relevant for the earthquake record, are available.




Integration  
 Dynamic analysis time domain

**Choose numerical integration scheme**  
 Implicit direct integration is applied over a specified time interval  
**Note: Standard HHT method is default**





Loading  
 Time domain  
 Integration  
 Earthquake



**Choose time series for ground acceleration**  
 9 recorded time series are available  
**Note: The chosen time series can be scaled**





Loading  
 Results  
 Earthquake response spectrum analysis  
 Run Analysis


**Specify the response spectrum to be used in the analysis**  
 The spectra available are all compatible with Eurocode 8

Integration  
 Run analysis


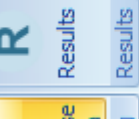

**Numerical integration**  
 Choose numerical integration method and parameters.  
 Implicit direct integration is applied over a specified time interval.  
 Note: Standard HHT method is default.






Loading  
 Time domain  
 Integration  
 Run  
 Run

**Earthquake**  
 Choose a time series of ground acceleration.  
 9 predefined time series are available, as well as all the time series that has been imported by the user.  
 Note: The chosen time series can be scaled.

Loading  
 Results  
 Run analysis  
 Run analysis

**Response spectra**  
 Specify the response spectrum to be used in the analysis.  
 The spectra available are all compatible with Eurocode 8.

 <p>Results</p>	<p>Earthquake response spectrum analysis</p> <p><b>Specify which and how results are presented</b> Results based on weighted response of all modes are default</p>
 <p>Run analysis</p>	<p><b>Results</b> Specify whether to present results based on response from one mode shape, or based on weighted response from all mode shapes.</p>

*Results ribbon page*

*Tooltips i modeling ribbon page har fått mer informasjon, bedre formatering og henvisninger til User's manual.*

 <p>Results</p>	<p>Earthquake response spectrum analysis</p> <p><b>Specify which and how results are presented</b> Results based on weighted response of all modes are default</p>	 <p>Results</p>	<p>Earthquake response spectrum analysis</p> <p><b>Results</b> Specify whether to present results based on response from one mode shape, or based on weighted response from all mode shapes.</p>
 <p>Results</p>	<p>Earthquake response spectrum analysis</p> <p><b>Specify which and how results are presented</b> Results based on weighted response of all modes are default</p>	 <p>Results</p>	<p>Earthquake response spectrum analysis</p> <p><b>Results</b> Specify whether to present results based on response from one mode shape, or based on weighted response from all mode shapes.</p>

**Σ F**  
Show resultants

Parameters Section  
Elastic Component  
Steel design

**Resultants of loading and reaction forces**

**Σ F**  
Show resultants

Parameters Section  
Elastic Component  
Steel design

**Show resultants**  
Shows the resultants of loading and reaction forces.  
Note: Numerical analyses methods will cause residuals.

Parameters Section  
Elastic Component  
Steel design

**Parameters**

Parameters Section  
Elastic Component  
Steel design

**Parameters**  
Show the national parameters, from Eurocode 8, used in steel design.

Parameters Section  
Elastic Component  
Steel design

Steel Design **Section**

Parameters Section  
Elastic Component  
Steel design

Steel Design **Section**  
Run a cross section steel design control.  
**More information:**  
User's manual section 6.10.

Parameters Section  
Elastic Component  
Steel design

**Run an elastic steel design control**

Parameters Section  
Elastic Component  
Steel design

**Elastic**  
Run an elastic steel design control.  
**More information:**  
User's manual section 6.10.

Parameters Section  
Elastic Component  
Steel design

Steel Design **Component**

Parameters Section  
Elastic Component  
Steel design

Steel Design **Component**  
Run a component steel design control.  
**More information:**  
User's manual section 6.10.

Active influence line  
u - x-displacement at joint 1  
Influence line

**Influence line for selected response parameter**  
Joint of response parameter is highlighted

Active influence line  
u - x-displacement at joint 1  
Influence line

**Active influence line**  
Select the influence line for the selected response parameter.  
The response parameter's joint will be highlighted in the model.



Active load train:

Number

**Select active load train**

Active load train:

Number

**Active load train**  
Select the active load train.

Number of load train positions:

Load train analysis

**Load train response is computed for the chosen number of equidistant positions of the load train on the load path**

Run and

Number of load train positions:

Load train analysis

**Number of load train positions**  
Load train response is computed for the chosen number of equidistant positions of the *load train* on the *load path*.

Max. value  Min. value

Run Xtrm analysis

**Algebraically maximum and minimum response due to the load train**

Max. value  Min. value

Run Xtrm analysis

**Max/Min value**  
Algebraically maximum and minimum response due to the load train.

Run Xtrm analysis

**Calculate worst case position of load train**  
One or more load trains must be defined in order to run this analysis.  
**Load trains may be defined at the "Loading"-page**

Run Xtrm analysis

**Run Xtrm analysis**  
Calculate the worst case position of the load train.  
One or more load trains must be defined in order to run this analysis.  
Load trains may be defined in the *Loading* ribbon.  
**More information:**  
**User's manual section 6.2.**

Run

Pause

Reset

Animation

**Start (or restart) the animation**

Run

Pause

Reset

Animation

**Run**  
Start (or restart) the animation.

Pause

Reset

Animation







**Pause the animation**

Pause

Reset

Animation

**Pause**  
Pause the animation.





 Reset  
 Move to start  
 Step back  
 Turn train  
 Step forward  
 Move end

**Reset the animation back to start position**

Active load train: Load train  
 Step Delay (ms): 200  
 Step size (# positions): 1

Load train control

**Delay between load train positions during animation**  
Reducing the delay will speed up the train

 Cycle previous  
 Maximum response  
 Cycle next  
 Convert to LC  
 LC name: LC\_Load train

**Move the train to a possible previous "maximum" position**  
The load train may cause the (same) maximum response for several positions





 Reset  
 Move to start  
 Step back

**Reset**  
Reset the animation back to start position.

Active load train: Load train  
 Step Delay (ms):  
 Step size (# positions):

Load train control

**Step delay**  
Delay between load train positions during animation.  
Reducing the delay will speed up the load train.

 Cycle previous  
 Maximum response  
 Cycle next  
 Convert to LC

**Cycle previous**  
Move the train to a possible previous maximum position.  
The load train may cause the (same) maximum response for several positions.

Active load train: Load train  
 Step Delay (ms): 200  
 Step size (# positions): 1




Load train control

**Choose active load train**

Active load train: Load train  
 Step Delay (ms): 200  
 Step size (# positions): 1

Load train control

**The load train is moved this number of load positions in each step**

 Maximum response  
 Cycle next  
 Convert to LC  
 LC name: LC\_Load train




**Move the train to (one of) the position(s) that causes maximum response**

Active load train: Load train  
 Step Delay (ms):  
 Step size (# positions):


**Active load train**  
Select the active load train.

Active load train: Load train  
 Step Delay (ms):  
 Step size (# positions):


**Step size**  
The load train is moved this number of load positions in each step.

 Maximum response  
 Cycle next  
 Convert to LC  
 LC name: LC\_Load train


**Maximum response**  
Move the load train to (one of) the position(s) that causes maximum response.


 LC name: LC\_Load train  
 Convert to LC


**Move the train forward to a possible "maximum position"**  
 The load train may cause the (same) maximum response for several positions.


 LC name: LC\_Load train  
 Convert to LC


**Cycle next**  
 Move the train forward to a possible maximum position.  
 The load train may cause the (same) maximum response for several positions.


 LC name: LC\_Load train  
 Convert to LC

**Convert current load train position to a load case (LC)**


 LC name: LC\_Load train  
 Convert to LC


**Convert to LC**  
 Convert current load train position to a load case.  
 Note: Do not convert load train steps where the loads in the step are close to any existing joints. The internal joints that will be created when converting may conflict with the existing ones.


 Show response parameters  
 Response parameter


**If relevant, show "time" variation of response parameters**


 Show response parameters  
 Response parameter

**Show response parameters**  
 If relevant, show a "time" variation of response parameters.


 Xtrm response  
 Xtrm response

**Determine maximum (absolute) responses (d, M, N, V) and the time of their occurrence**


 Xtrm response  
 Xtrm response

**Xtrm response**  
 Determine maximum (absolute) responses (d, M, N, V) and the time of their occurrence.

Amplitude  
 Dynamic amplification factors  
 Phase angle  
 Change result

**Amplitude values of "steady state", harmonic response**

Amplitude  
 Dynamic amplification factors  
 Phase angle  
 Change result

**Amplitude**  
 Show amplitude values of "steady state", harmonic response in the result diagram.

Amplitude  
 Dynamic amplification factors  
 Phase angle  
 Change result

**Dynamic amplification factors for "steady state", harmonic response**

Amplitude  
 Dynamic amplification factors  
 Phase angle  
 Change result

**Dynamic amplification factors**  
 Show dynamic amplification factors for "steady state", harmonic response in the result diagram.

<p>Amplitude</p> <p>Dynamic amplification factors</p> <p>Phase angle</p> <p>Change result</p> <p><b>Phase angles of individual "steady state" responses</b></p>	<p>Amplitude</p> <p>Dynamic amplification factors</p> <p>Phase angle</p> <p>Change result</p> <p><b>Phase angle</b></p> <p>Show phase angles of individual "steady state" responses in the result diagram.</p>	<p>Show response parameters</p> <p>Response parameter</p> <p>Show peak estimates</p> <p>Peak estimates</p> <p><b>For any response parameter, show the response in each modal shape</b></p>	<p>Show response parameters</p> <p>Response parameter</p> <p>Show peak estimates</p> <p>Peak estimates</p> <p><b>Show response parameters</b></p> <p>For any response parameter, show the response in each modal shape.</p>
<p>Show peak estimates</p> <p>Peak estimates</p> <p><b>Show peak estimates</b></p> <p>For any response parameter, show the peak response values estimated by different weighting procedures.</p>	<p>Show peak estimates</p> <p>Peak estimates</p> <p><b>Show peak estimates</b></p> <p>For any response parameter, show the peak response values estimated by different weighting procedures.</p>		

### Settings

I settings-dialogen har det ikke vært innført et eneste *tooltip* før det ble gjort i **faq2D 3.1**. Det er vanskelig å forstå hvordan alle *settings* fungerer, spesielt siden det er så lite informasjon tilgjengelig. Dessuten har mange av innstillingene i dialogboksen ikke vært i bruk i det hele tatt, og burde vært fjernet tidligere. Derfor er mange *settings* fjernet i **faq2D 3.1**. En del *settings* har blitt lagt til i takt med endringer og nye implementasjoner som er gjennomført.


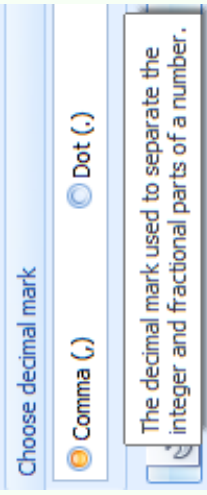
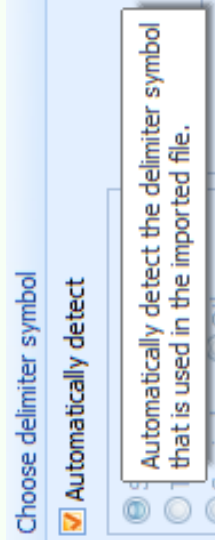
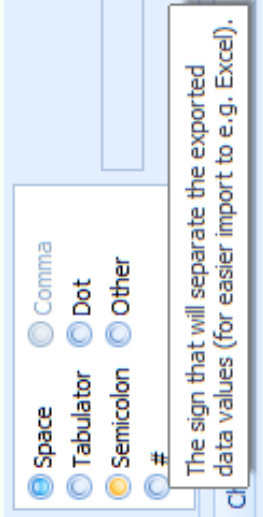


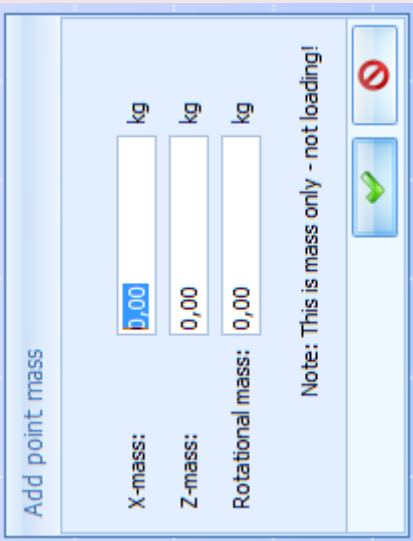

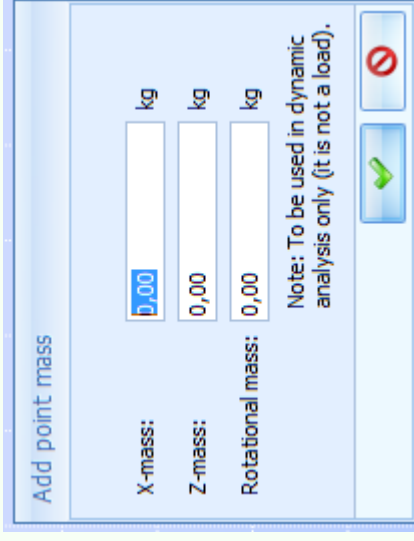
<p><b>Background colors</b></p> <p>Top color: <input type="text" value="234; 234; 255"/></p> <p>Bottom color: <input type="text" value="155; 188; 255"/></p> <p><b>Grid and Grid:</b></p> <p>The background colors are interpolated between the top and bottom colors.</p> <p>Second split value: <input type="text" value="100,00 %"/></p> <p>Second split level for steel design utilization results. Currently not editable.</p>	<p><b>Result colors</b></p> <p>Positive section force: <input type="text" value="255; 0; 0"/></p> <p>Negative section force: <input type="text" value="0; 0; 255"/></p> <p>First split level: <input type="text" value=""/></p> <p>Second split level: <input type="text" value=""/></p> <p>Color that indicates a positive section force.</p> <p>Color that indicates a negative section force.</p> <p><b>Scaling of result diagrams*:</b> <input type="text" value="10,00 %"/></p> <p><input checked="" type="checkbox"/> Zoom to model</p> <p><input checked="" type="checkbox"/> Use custom</p> <p>Scale of result diagrams compared to the characteristic length in the model.</p> <p>*The scale will not be changed until a new analysis is performed.</p>	<p><b>Result colors</b></p> <p>Positive section force: <input type="text" value="255; 0; 0"/></p> <p>Negative section force: <input type="text" value="0; 0; 255"/></p> <p>First split level: <input type="text" value=""/></p> <p>Second split level: <input type="text" value=""/></p> <p>Color that indicates a negative section force.</p> <p><input checked="" type="checkbox"/> Zoom to model when displaying results.</p> <p><input checked="" type="checkbox"/> Use custom mouse pointer.</p> <p>Zoom to model when displaying result diagrams, after a successful analysis has been performed.</p>	<p><input checked="" type="checkbox"/> Remove numerical noise.</p> <p>Remove numerical noise in dynamic analyses by default.</p> <p>Default shift value: <input type="text" value="0"/></p> <p>Decides where on the frequency axis the eigenvalues (for free vibration and modal analysis) should be extracted.</p>
---	--	---	---

<p>First split level: <input type="text" value="50,00 %"/></p> <p>Second split value: <input type="text" value="100,00 %"/></p> <p>First split level for steel design utilization results. Currently not editable.</p>	<p><input checked="" type="checkbox"/> Merge overlapping joints on default.</p> <p><input type="checkbox"/> <i>Include own weight on default if no external loading</i> Merge overlapping joints on default, by deleting the joint (2) that was placed upon an existing joint (1). If any members are connected to joint 2, they will now become connected to joint 1. None of the properties of joint 2 (hinge, rotational spring etc.) will be transferred to joint 1.</p>	<p><input checked="" type="checkbox"/> Show arch member height warning</p> <p>Show a warning dialog if the height/radius of an arch member may cause numerical instabilities.</p>	<p>Default eigenvalue algorithm: Subspace iteration</p> <p>Default: Algorithm used for solving eigenvalue problems (free vibration, modal analysis and linearized buckling).</p>
<p>Print flag: <input type="text" value="1"/></p> <p>This value controls how much information that is written to the log file.</p>			<p>Default eigenvalue algorithm: Subspace iteration</p> <p>Default: Algorithm used for solving eigenvalue problems (free vibration, modal analysis and linearized buckling).</p>

<p><input checked="" type="checkbox"/> Include own weight on default if no external loading is applied in analysis.</p> <p><input type="checkbox"/> <i>Show element size warning</i> The own weight of the structure will be added to the default (active) load combination when no external loading has been applied before running an analysis.</p>	<p>Element selection offset: <input type="text" value="15"/></p> <p>Selection radius (other items): <input type="text" value="10"/></p> <p>The selection offset indicates how close to an element you must click to mark it as selected. The default value is 15 pixels. Too low values make it difficult to select elements. Too high values may make it difficult to select other items.</p>	
<p>Selection radius (other items): <input type="text" value="10"/></p> <p>The selection radius indicates how close to a structural item you must click to mark it as selected. Affected items and dialogs: Loads, coupling spring dialog, eccentricity dialog, joint dialog, release dialog.</p>	<p>Selection radius (other items): <input type="text" value="10"/></p> <p>The selection radius indicates how close to a structural item you must click to mark it as selected. Affected items and dialogs: Loads, coupling spring dialog, eccentricity dialog, joint dialog, release dialog.</p>	
<p><input checked="" type="checkbox"/> Show element size warning</p> <p>Show a warning dialog if the elements in the mesh may cause numerical problems.</p>	<p>Selection buffer size: <input type="text" value="16384"/></p> <p>Size of graphics memory buffer in bytes. The default value should be sufficient for the OpenGL graphics performance. Do not set a lower value than 128 bytes.</p>	

## Diverse dialogbokser

<p><u>Export to text file</u></p> <p>I den nye dialogboksen for eksport er det lagt til <i>tooltips</i> for de fleste innstillingene.</p>	 <p><b>Joints</b> Members: Joints: ID and coordinates.</p>	 <p><b>Members</b> Members: ID, type, geometry, height, joints, material and cross sections.</p>	 <p><b>Cross sections</b> Cross sections: Name, type, A, I, h1 and h2.</p>
 <p><b>Results for members</b></p>	 <p><b>Results for members</b> Cross sectional forces for all elements in the selected members, for the last active analysis.</p>	 <p><b>All members</b> Select members Select all members or select a group of members based on their ID.</p>	 <p><b>Comma (,)</b> Dot (.) Choose decimal mark The decimal mark used to separate the integer and fractional parts of a number.</p>
 <p><b>Results for members</b> The sign that will separate the exported data values (for easier import to e.g. Excel).</p>	 <p><b>Space</b> Comma Semicolon Dot # Other Choose delimiter symbol</p>	 <p><b>Renumber IDs</b> Renumber the joint and member IDs before export. Note: If exporting "Results for members" and using the option "Select members", renumbering should be done in the toolbox before exporting.</p>	 <p><b>Write to separate files</b> Renumber IDs Export joints, members, cross sections and results to individual files.</p>

<p><u>Import joints</u></p> 	<p>I den nye dialogboksen for import er det lagt til <i>tooltips</i> for de fleste innstillingene.</p> 	<p><u>Add point mass</u></p>
		<p>Beskjeden i <i>Note</i> er endret i hht. endring av <i>tooltips</i> til knappen som åpner <i>Add point mass</i>-dialogen.</p>
		
		

## C.3 GridView events

# GridView events

General rules for `DevExpress.XtraGrid.Views.Grid.GridView` events which are used in the `GridView` controls that display *joint* and *member data* in the *left panel* of **fap2D**.

### Remark

Only the events that have been used in versions **3.0.0 - Beta** and **3.1**, are covered in this document.

### Additional information

Search for “GridView events” in the DevExpress online documentation:  
<https://documentation.devexpress.com>

### Events

fap2D has, during development, used the following `GridView` events:

<code>RowCellStyle</code>	Fired when hovering over the <code>GridView</code> and at every other changes made to the <code>GridView</code> (in reality it happens continuously). Used to control the layout and formatting of the rows in the <code>GridView</code> .
<code>SelectionChanged</code>	Registers which cell(s) are currently selected in the <code>GridView</code> .
<code>FocusedRowChanged</code>	Fired at every change to a cell, even when the action causing the cell to change is performed outside the <code>GridView</code> .
<code>LostFocus</code>	Fired when the <code>GridView</code> loses focus. Because DevExpress encapsulates Windows Forms controls this event will happen twice every time (the DevExpress encapsulation will lose focus to the Windows Forms base class and regain focus afterwards). Therefore it happens every time the user performs a mouse action as long as the <code>GridView</code> has got focus.
<code>CellValueChanged</code>	Fired after a cell has been edited, before navigating to another cell or losing focus.
<code>CellValueChanging</code>	Fired for every time an input value (character) is added or removed to a cell. Fired before <code>CellValueChanged</code> .
<code>ValidateRow</code>	Fired every time one row loses focus, before another row gains focus. Used to validate values in the row.



### Application specific information

The property `IsFocusedView` is important to use in the events in order to avoid unnecessary work being done in then GridView when it hasn't got focus. This is because the GridViews in `fap2D` are connected to the actions in the *modelling panel* so as to update the structural data in the tables when *joints* and *members* are being edited by the use of mouse, and vice versa.

Events that are fired only when the GridView has got focus:

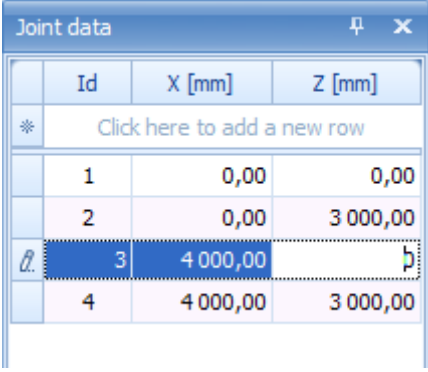
- `CellValueChanging`
- `CellValueChanged`
- `LostFocus`

Events that are fired even when the GridView hasn't got focus:

- `RowCellStyle`
- `FocusedRowChanged`
- `SelectionChanged`

Listed below are some important event scenarios, which occur in the `fap2D` implementation of the GridView for the *joint data* table. Some of the scenarios are application specific because some of the events are programmed to fire other events.

The documentation of the GridView events is comprehensive and complex, so we have established the order of the events purely by debugging `fap2D`.



	Id	X [mm]	Z [mm]
*	Click here to add a new row		
	1	0,00	0,00
	2	0,00	3 000,00
⌚	3	4 000,00	
	4	4 000,00	3 000,00

*Joint data* table in `fap2D`.

## Scenarios

### Hovering over a GridView when it **hasn't** got focus

1. RowCellStyle

### Clicking at a GridView (cell) when it **hasn't** got focus

1. FocusedRowChanged
2. SelectionChanged
1. LostFocus
  - 1.1. SelectionChanged, because the property view.ClearSelection is used.
  - 1.2. CellValueChanged, because the property view.SetRowCellValue is used.

### Using the mouse pointer to change the selected cell when the GridView **has** got focus

1. LostFocus
2. CellValueChanged
3. ValidateRow, if selecting a cell in another row.
4. FocusedRowChanged
5. SelectionChanged
6. LostFocus
  - 6.1. SelectionChanged, because the property view.ClearSelection is used.
  - 6.2. CellValueChanged, because the property view.SetRowCellValue is used.

### Using TAB or ENTER to change the selected column **within the same row** in the GridView

1. None!

### Using TAB or ENTER to change the selected row in the GridView

1. ValidateRow
2. FocusedRowChanged
3. SelectionChanged
4. LostFocus
  - 4.1. SelectionChanged, because the property view.ClearSelection is used.
  - 4.2. CellValueChanged, because the property view.SetRowCellValue is used.

### Using the mouse pointer to leave the GridView

1. LostFocus
  - 1.1. CellValueChanged, because the property view.SetRowCellValue is used.
2. ValidateRow

## C.4 Mouse events

# Mouse Events

General rules for mouse events in the *modeling panel* (`fap2D.Gui.FrmModel.cs`).

**Remark:** Drag-and-drop operations are not used

### ***Additional information:***

[Mouse Events in Windows Forms](#)

### **Usage in fap2D:**

fap2D utilizes the following mouse events:

- MouseDown
- MouseClick
- MouseMove
- MouseUp
- MouseHover
- MouseLeave
- MouseWheel

According to the Windows Forms Documentation the mouse events happens in the following order:

1. MouseEnter
2. MouseMove
3. MouseHover/MouseDown/MouseWheel
4. MouseUp
5. MouseLeave

and in the following order for click:

1. MouseDown
2. (Click) – Click event on a WinForm control
3. MouseClick
4. MouseUp

**Remark:** The events of a mouse click are part of a *mouse (click) sequence*, that is, they are step 3 and 4 in the mouse event sequence.

### ***Position/Coordinates:***

The mouse positions is sent as event arguments. The position is in *screen coordinates*(/pixels). To use the mouse position it must first be converted to *world coordinates*.

Note that the *screen coordinates* should only be used when calculating the distance moved, or when converting to *world coordinates*. *Screen coordinates* do not change along with zoom or translation of the *world coordinates* system

## Mouse Move

Special care must be taken if the mouse is moved during a mouse click sequence.

*Minimum distance* of a mouse movement is a number of pixels (current value: 28)

## Mouse Actions:

Enum for different mouse actions

```
public enum MouseAction
```

Description	Value	Description	Value	Description	Value
Take no action or select drawable object.	None	Zoom to the selected rectangle.	ZoomRectangle	Concentrated damper.	Concentrated-Damper
Move the model or create new member.	Member	X-displacement release - release local X-axis.	XDisplacement Release,	Load path	LoadPath
Create a new joint.	Joint	Rotation release.	Rotation-Release	Initial strain or temperatur.	InitialStrain Temp
Create a new support.	Support	Prescribed displacement.	Prescribed-Displacement	Include local imperfection.	IncludeLocal-Imperfection
Create a new concentrated load.	PointLoad	Point mass.	PointMass	Hand.	Hand
Create a new distributed load.	LineLoad	Spring.	Spring	While doing analysis	Loading

## Properties

Some important and useful properties which needs to be kept track of are:

- mouse moved
  - relative to the origin of a mouse click sequence
- mouse moved minimum distance
  - relative to the origin of a mouse click sequence
- snap point
- the drawable item clicked
  - the one (of potentially many) item clicked in the mouse click sequence
- selected item
- [current/last mouse coordinates (sent as argument)]
- first mouse coordinates
  - the origin of the mouse click sequence
- mouse pressed
- [left mouse pressed]

## MouseMove Event

Mouse event raised when the mouse is moved inside the modeling panel.

**Remark:** this event is called a preset number of times per second.

### *Properties to Update*

- current/last mouse coordinates → update to argument values
- mouse pressed → `false`

### *Actions for button(s) clicked:*

#### **All Buttons:**

Reset mouseHover timer.

Assert if mouse has moved minimum distance.

#### **Left Mouse Button**

- If in results tab: don't do anything (`Mouse.MouseAction.None`)
- If not in results tab: either move selected items to this temporary position or update the size of the selection box (`Mouse.MouseAction.None`)
- Pan camera (`Mouse.MouseAction.Hand`)
- Update the position of a member in creation (`Mouse.MouseAction.Member`)
- Update the size of the selection box of the zoom rectangle (`Mouse.MouseAction.ZoomRectangle`)

#### **Middle Mouse Button**

Pan camera.

## MouseLeave Event

Event when the mouse leaves the modeling panel/window.

Refresh the modeling panel/window.

## MouseDown Event

First event when any mouse button is pressed.

### *Properties to Update*

- mouse moved → `false`
- mouse moved minimum distance → `false`
- snap point → `null`
  - if MouseBtnL or MouseBtnR → find snap point
- the drawable object clicked → `null`
  - get drawable object
- selected objects → `null`
- current/last mouse coordinates → update to argument values
- first mouse coordinates → update to argument values
- mouse pressed → `true` (not for MouseBtnR)

### *Actions for button(s) clicked:*

#### **All Buttons:**

Mouse button settings are initially changed.

#### **Left Button:**

- Do nothing (`Mouse.MouseAction.None`)
- Select/reselect one or more drawableObjects (`Mouse.MouseAction.None`)
- Start creating a new member (`Mouse.MouseAction.Member`)

**Remark:** New members has to be completed in a MouseUp event.

#### **Right Button:**

Select an element in the result tab.

## MouseClicked Event

Mouse event when a mouse button is clicked. This event is raised after the MouseDown event.

### *Actions for button(s) clicked:*

#### All Buttons:

-

#### Left Button:

- Do nothing ([Mouse.MouseAction.None](#))
- Create a new joint or internal joint ([Mouse.MouseAction.Joint](#))
- Create/replace a support on a joint ([Mouse.MouseAction.Support](#))
- Create/replace a point load on a joint or a submember ([Mouse.MouseAction.PointLoad](#))
- Create/replace a line load on a submember ([Mouse.MouseAction.LineLoad](#))
- Create a rotation release on a joint with multiple connecting members or on an internal joint ([Mouse.MouseAction.RotationRelease](#))
  - and display relevant dialog box ([FrmReleaseDialog](#))
- Create an axial release on a joint with multiple connecting members or on an internal joint ([Mouse.MouseAction.XDisplacementRelease](#))
  - and display relevant dialog box ([FrmReleaseDialog](#))
- Create a prescribed displacement on a joint with a support ([Mouse.MouseAction.PrescribedDisplacement](#))
  - and display relevant dialog box ([FrmPrescribedDisplacementDialog](#))
- Create initial strain or temperature on a submember ([Mouse.MouseAction.InitialStrainTemp](#)),
  - and display relevant dialog box ([FrmTempStrainIndividualMembers](#))
- Create/replace a local imperfection on a submember ([Mouse.MouseAction.IncludeLocalImperfection](#))
- Create a point mass on a joint ([Mouse.MouseAction.PointMass](#))
- Create/replace a boundary spring or a coupling spring on a joint ([Mouse.MouseAction.Spring](#))
- Create a concentrated damper on a joint ([Mouse.MouseAction.ConcentratedDamper](#))
- Create a load path on a submember ([Mouse.MouseAction.LoadPath](#))

**Remark:** Spring and concentrated damper displays a dialog box before any mouse actions.

#### Right Button:

Find nearby objects and show right click menu if the user is in the model tab, or show dialogs displaying the results for each elements if the user is in the results tab.

## MouseUp Event

Last event when a mouse button is pressed. This event is raised after the MouseDown and MouseClicked event.

### *Properties to Update*

- current/last mouse coordinates → update to argument values
- mouse pressed → `false`
- (left mouse pressed) → `false`
- snap point → `null`
- drawable object/item clicked → `null`
- selected objects/items → `null`
- selection area → `Rectangle.Empty`
- start positions → `null`

### *Actions for button(s) clicked:*

#### All Buttons:

-

#### Left Button:

- Do nothing (`Mouse.MouseAction.None`)
- Move every selected structural item (`Mouse.MouseAction.None`)
- Finish creating a new member (`Mouse.MouseAction.Member`)
- Zoom into a zoom rectangle (`Mouse.MouseAction.ZoomRectangle`)

#### Right Button:

-



## MouseHover Event

Event when the mouse is resting at a position for a given period.

This event may be fired at any stage of a mouse click sequence, i.e. before, between and after MouseDown and MouseUp event.

Reset mouseHover timer (e.g. when mouse moves):

```
MouseInput.ResetMouseHover(openGLControl.Handle);
```

### ***Actions in Active Tab:***

#### **Modelling/Load/Analysis Tab**

- Mousover object/item → one of the submembers which contains the mouse pointer at the moment.

**Remark:** When creating a new member, then member may have coinciding joints, which may throw an exception.

#### **Results Tab**

- Mouse over object/item → closest load train

### ***Tool tip***

Display relevant mouse over text.

## MouseWheel Event

Event when the mouse scroll wheel is used.

Remove marked hovering objects:

```
_mouseOverObject.IsMouseOver → false
```

Reset timer.

Zoom the camera a given number of steps (*detents*) the mouse wheel has rotated.

## C.5 Testing av analyser (*tutorials*)

Enkel testing av øvingsoppgaver.

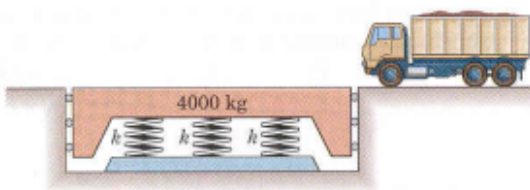
# Testing av dynamiske analyser i fap2D

TKT4122 Mekanikk 2, høst 2010

Faglærer: Kjell H. Holthe

### Øving 11 - Oppgave 1

#### OPPGAVE 1



Figuren viser en prinsippskisse av ei vekt for tyngre kjøretøy. Vekta består av en plattform med masse  $m = 4000$  kg. Denne er understøttet av tre identiske fjærer, som hver har stivhet  $k$ . Det er ønskelig at egenfrekvensen for fri, vertikal svingning av det ubelastede systemet (intet kjøretøy på vekta) ikke skal overskride 3 Hz. Sett  $g = 9.8$  m/s<sup>2</sup>.

- Bestem maksimal verdi av fjærkonstanten  $k$  slik at kravet til egenfrekvens er tilfredsstillt.
- Hva blir egenfrekvensen (i Hz) til systemet når lastebilen med masse  $M_{bil} = 12\,000$  kg har kjørt opp på vekta?
- Hva blir statisk vertikalforskyvning av vektas plattform når lastebilen kjører på vekta?

Svarantydning:  $k_{max} = 474$  kN/m       $f_e = 1.5$  Hz       $x_{stat} = 83$  mm

#### Modell i fap2D

Plattformen er modellert som en *beam member*:

Lengde,  $L = 2\,000$  mm.

Tverrsnittsareal,  $A = 1$  m<sup>2</sup>.

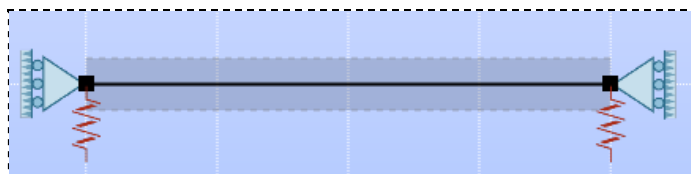
2. arealmoment,  $I = 100\,000\,000$  mm<sup>4</sup>.

Materiale:  $E = 100\,000$  MPa,  $d = 2\,000$  kg/m<sup>3</sup>.

Masse,  $m = 1 \times 2 \times 2\,000 = 4\,000$  kg.

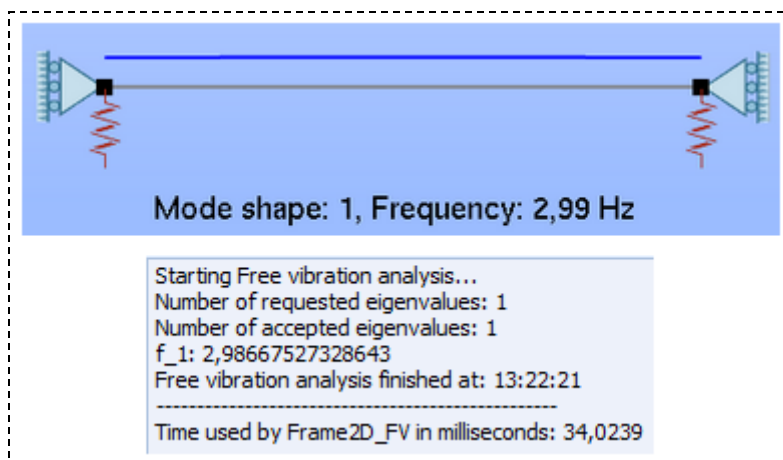
Fjærstivhet:

To fjærer med  $k = 711$  kN/m (tilsvarer tre fjærer med  $k = 474$  kN/m).



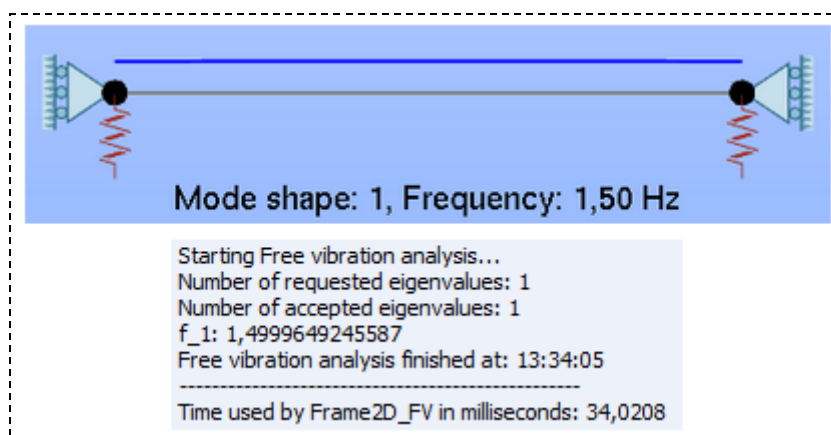
### 1a) Kontroll av fjærstivhet - Free vibration analysis:

Som figurene under viser så holder vi oss akkurat innenfor kravet om maksimum 3 Hz. Vi aksepterer at svaret vårt ikke er akkurat 3 Hz siden det er umulig å få kjørt en analyse der plattformen er 100 % stiv.



### 1b) Egenfrekvens med bil - Free vibration analysis:

Vi legger på et internt joint med en punktmasse på 12 000 kg og kjører ny analyse. Resultatet stemmer overens med svarantydningen, som figuren under viser.



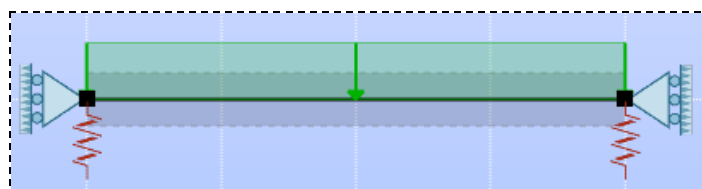
### 1c) Statisk vertikalforskyvning - Linear static analysis:

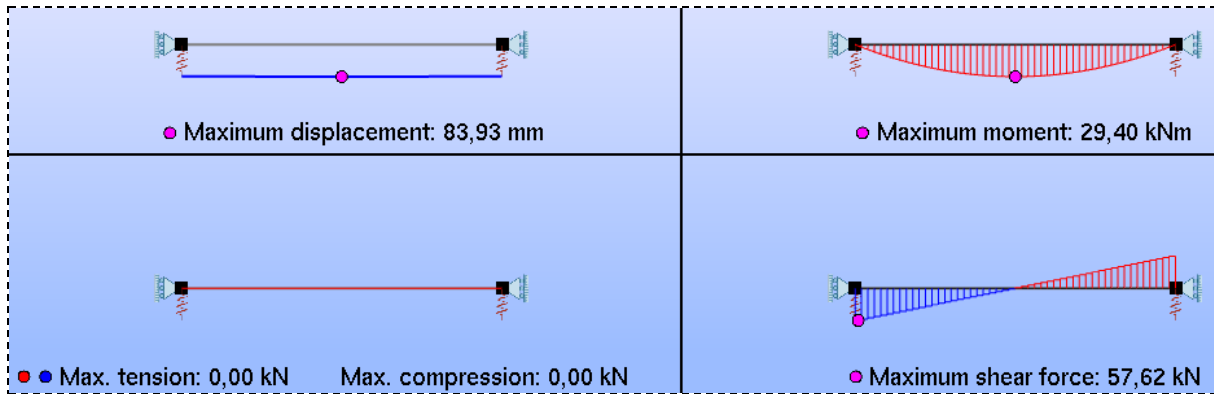
Siden *point mass* kun fungerer som en dynamisk masse i **fap2D** fjerner vi dem og legger heller på en fordelt last for å simulere bilen:

Total last, bil:  $F = 12\,000\text{ kg} \times 9,8\text{ m/s}^2 = 117,6\text{ kN}$ .

Fordelt last:  $F/L = 117,6\text{ kN} / 2\text{ m} = 58,8\text{ kN/m}$ .

Bidraget fra den fordelte lasten fører til samme nedbøyning som i svarantydningen:

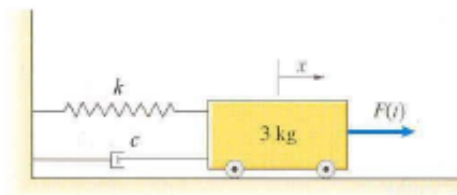




### Øving 13 - Oppgave 1

Analysetype: *Free vibration og Dynamic time/frequency domain (forced vibration).*

#### OPPGAVE 1



Figuren en idealisert, dempet enfrihetsgradssvinger som er påkjent av en harmonisk belastning

$$F(t) = F_0 \cos(\omega t)$$

Data:  $k = 12 \text{ N/m}$      $c = 1.2 \text{ Ns/m}$      $F_0 = 6 \text{ N}$

- Bestem egenperioden  $T_e$  og dempningsforholdet  $\zeta$ .
- Bestem forskyvningen  $x_{stat}$  for statisk last ( $\omega = 0$ ).
- Bestem responsamplituden  $x_{max}$  ved resonans.
- Det er ønskelig at amplituden  $x_{max} < 0.4 \text{ m}$ . Bestem minimum lastfrekvens  $\omega$  som sikrer dette.

Svarantydning:

- $T_e = 3.14 \text{ s}$      $\zeta = 0.1$
- $x_{stat} = 0.5 \text{ m}$
- $x_{max} = 2.5 \text{ m}$
- $\omega > 2.976 \text{ rad/s}$

#### Modell i fap2D

I fap2D brukes en modell som er skalert opp med en faktor 1000, for enklere tilpasning til enhetene i programmet. Dessuten er modellen speilvendt, også dette fordi det er mest lettvisst. Visuelt ser det ut som punktdemperen er vertikal, men det er bare på grunn av hvordan symbolet tegnes på modellen.

Skalerte verdier:

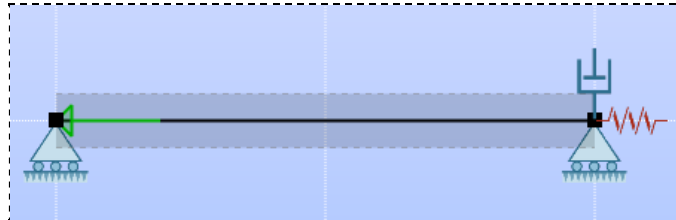
- $m = 3000 \text{ kg}$ ,  $k = 12 \text{ kN/m}$ ,  $c = 1,2 \text{ kNs/m}$ ,  $F_0 = 6 \text{ kN}$ .

Modell av masse:

- Member type: *Beam*

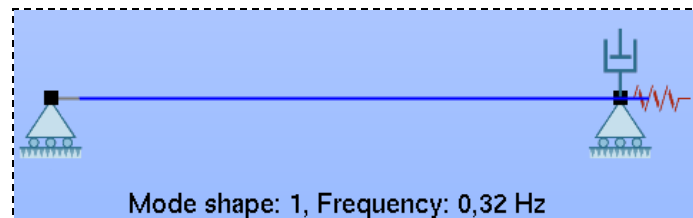
- Lengde: 1 m
- Tverrsnitt: 0,01 m<sup>2</sup>
- Materiale: E = 1 000 000 MPa, massetetthet, d = 300 000 kg/m<sup>3</sup>

Massen er modellert som en tilnærmet stiv bjelke som tilsammen veier 3 000 kg.



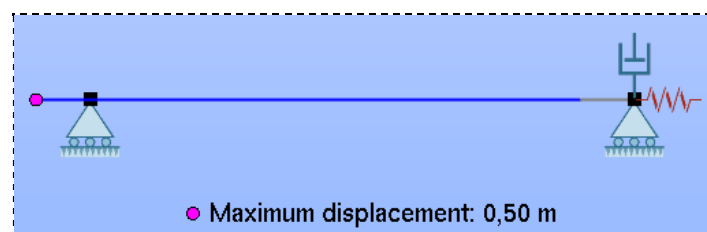
1a) Free vibration analysis:

En egenfrekvens på tilnærmet 0,318 Hz. Det tilsvarer samme egenperiode, T = 3,14 s.



```
Starting Free vibration analysis...
Number of requested eigenvalues: 1
Number of accepted eigenvalues: 1
f_1: 0,318309824956706
Free vibration analysis finished at: 18:43:08
Time used by Frame2D_FV in milliseconds: 37,023
```

1b) Linear static analysis: Samme resultat som i svarantydningen.



# Tillegg D

## Logg

### D.1 Logg - Torjus Sandviken

#### Logg - Torjus Sandviken

Loggen viser progresjonen i arbeidet med masteroppgaven, ved å vise hva som har blitt gjort hver dag. Loggen er i hovedsak ment for eget bruk. Derfor presenteres bare ukentlige oppsummeringer fra loggen i dette dokumentet. Detaljer om endringer i kildekoden presenteres ikke her, bare kommentarer til endringene i hht. punktene i *Release notes*.

##### Før offisiell start (Kalenderuke 52 - 2) - Oppstart

Møte m/veileder, 8. januar:

- 1. møte ble gjennomført for å komme tidlig i gang med planleggingen.
- Bestemte oss for å fortsette med ukentlige møter, slik som i prosjektoppgaven.
- Fikk tilbake prosjektrapporten fra før jul og ser gjennom den før neste møte.
- Planlegging: Dynamisk analyser forskyves til lenger utpå våren, konstruksjonsmodell og *GUI* prioriteres først. Feil knyttet til *arch members* og *aksial-* og *skjærkraft-* diagrammer blir første oppgave for oss.

Arbeid:

- Innførte *DevExpress 13.2.5*-oppdatering i kildekoden. Førte ikke til spesielle endringer.

##### 1. uke (Kalenderuke 3) - Planlegging, betatesting, bug system

Møte m/veileder, 14. januar:

- Leverte masterkontrakt til Kolbein. Bestemte oss for å levere felles besvarelse.
- Betatesting: Diskuterte om **fap2D** skal brukes i øvinger i faget *KMEK*. Faglærer er positiv, men programmet må testes på Windows 8 før vi kan ta en avgjørelse.
- Vi påpekte at det er mangel på dokumentasjon av kode og programmet generelt. Det bør lages mer dokumentasjon slik at fremtidige utviklere får flere holdepunkter.

Arbeid:

- Planlegging:
  - Satte av datoer til innledende oppgaver, blant annet oppretting av system for feilretting og prioritering av arbeidsoppgaver innen feilretting.
  - Bestemte at *tooltips* skal være ferdig gjennomgått innen 28. februar.
- Testet installasjon og kjøring av **fap2D** på Windows 8 med hjelp fra medstudentene Henrik Thorsen og Torger Nordgård, siden vi selv hadde Windows 7 selv. Ingen problemer under testing, ergo kan betatesting i *KMEK* gjennomføres.
- Innførte *DevExpress 13.2.6*-oppdatering i kildekoden til **fap2D**. Skapte noen problemer i kildekoden som tok tid å rette (referanser til *DevExpress* ble slettet). Måtte også gjøre endringer pga. endring av deler av rammeverket.
- Feil/*bug*-system: Prøvde å finne et bedre system for å holde oversikt over feil/*bugs* i programmet. Per nå brukes et *Excel*-ark, uten versjonskontroll. *Team Foundation Server* er det enkleste alternativet, men vil ikke kunne kobles til kildekoden vår (Som bruker SVN)

- Espen rettet deler av feilen i *aksial-* og *skjærkraft-* diagrammer slik at de tegnes på riktig side av *members*.

## 2. uke - Planlegging, betatesting, feilretting, bug system

Møte m/veileder, 21. januar:

- Diskuterte diverse feilrettinger:
  - *OpenGL*-problemer (tegning av tekst).
  - Overlappende *members*.
  - *Arch members* (validering av geometri og deling i *submembers*),
  - *Aksial-* og *skjærkraft-*diagrammer (vilkårlig tegning for grenseverdi-vinkler).
- Kolbein poengterer at eksport/import bør implementeres.

Arbeid:

- Demo av **fap2D** ved Kolbein i *KMEK*. Vi presenterte samtidig opplegget for betatesting til studentene, med feilrapportering, *support* og evaluering.
- Feilretting og endringer:
  - *Aksial-* og *skjærkraft-*diagrammer: Jobbet sammen med Espen. Brukte mye tid, men slet med å komme videre i rettingen.
  - *Arch members*: Jobbet videre med validering av geometri og deling i *submembers*. Komplisert arbeid, tidkrevende.
  - *Tidsstempler* i *error list* og *information list*.
- Feil/bug-system: Prøvde å integrere et system kalt **Beurtle** med versjonskontrollen for kildekoden. Var komplisert så jeg droppet å se mer på det.

## 3. uke - Betatesting, feilretting, bug system

Møte m/veileder - *Ikke behov for møte, møttes to ganger forrige uke*

Arbeid:

- Feilretting:
  - *Arch members*: Fortsatte med de samme feilene.
  - Registrert flere nye feil, blant annet ved tegning av *joints* og forenkling av brukerinteraksjon.
  - Prioriterte en del småfeil, for å få unna noen punkter på listen over feil.
- Betatesting:
  - La ut skjema for feilretting på fagets hjemmeside.
  - Kolbein informerte om problemer for studentene i *KMEK* med å kjøre **fap2D** på datasal-PC-er fordi loggfilen skrives til mappe der studentene ikke har skriverettigheter. Umulig å få kjørt analyser. Sendte mail til Kenneth Sundli som har ansvaret for installeringen på datasal for å gi skriverettigheter til studentene. Ble i orden i løpet av helgen.
  - Svart på mange henvendelser denne uken, har tatt mye tid, men er viktig å prioritere.
- Feil/bug-system: Valgte at *Team Foundation Server* skal brukes siden det er enklest. Får ikke integrert systemet med versjonskontrollen for kildekoden så det blir tungvint, men får duge.

## 4. uke - Feilretting, nye implementasjoner

Møte m/veileder, 3. februar: *Var syk, fikk referat fra Kolbein. Ingen store saker.*

Arbeid:

- Jobbet med enkle implementasjoner og rettinger pga. sykdom. Blant annet formatering av output, småfeil i *GUI* og bruk av ulike farger for *bar*, *cable* og *strut members*.
- Endret rekkefølgen på knappene til *structural items* i *Modeling ribbon page*. De mest brukte knappene puttes lengst til venstre i *ribbon*.
- Overlappende *joints*: Brukt veldig mye tid på å finne ut hvordan flytting av *joints* og behandling av overlappende *joints* skal foregå. Mye tankearbeid og forstudium.

### 5. uke - Feilretting, nye implementasjoner, ny versjon

Møte m/veileder, 11. februar:

- La fram noen forslag til endringer for å få tilbakemelding fra Kolbein.
- Diskuterte implementasjon av en nedre grense for elementlengde. Nyttig for å varsle brukere om at for korte elementer ikke er hensiktsmessig.
- Diskuterte hvordan *joints* bør oppføre seg når de flyttes. Komplisert å koble flyttende *joints* til *members*.
- Dersom brukeren ikke oppretter laster i modellen bør ikke analysene automatisk legge til egenvekt. Vi er enig om at brukeren kan bli lurt og skal endre dette.

Arbeid:

- Gått gjennom forslag til oppgavetekst som Kolbein sendte til Espen og meg. Teksten er litt vag så vi vil ha inn noen presiseringer.
- Feilretting:
  - Overlappende *joints*. Komplisert, brukt mye tid.
  - Registrert en del nye feil.
  - Slettede modellfiler: Fjernes nå fra *application menu*.
  - *Members* med *joints* i samme punkt.
  - Jobbet med validering av input og formatering av input/output til/fra dialogbokser.
- Innførte *DevExpress 13.2.7*-oppdatering i kildekoden. Førte ikke til spesielle endringer.
- Installerte en ny versjon av **fap2D** som Espen laget i går. Ser ut til at de nye endringene fungerer bra.

### 6. uke - Feilretting, endringer, dokumentasjon av endringer (release notes)

Møte m/veileder, 19. februar:

- Tegning av *arch members*: Tegnes for hakkete, bestemmer oss for å tegne dem glattere. I resultatvinduet skal *arch members* derimot tegnes med de rette linjene som er definert av elementene i *member*-en.
- Kolbein synes det er tungvint å måtte deaktivere knappene for *loads/draw member* før man kan aktivere dem igjen. Vi skal endre dette så brukeren kan benytte bare ett tastetrykk i stedet for to.
- *DevExpress*: Diskuterte om vi bør fornye lisensen på dette rammeverket, slik at vi fortsatt får oppdateringer. Virker som rammeverket fungerer stabilt og godt nå, så det er ikke vits å fornye lisensen.
- Diskuterte begrensningene vi har pga. tungvint, manuelt feil/*bug*-system. Ingen god løsning så vi må bare fortsette med det samme som før.

Arbeid:

- Jobbet mye med dokumentasjon av alle endringene som er gjort slik at vi kan inkludere dem i *release notes* for neste versjon. Gjør det enkelt for brukere å se hva og hvorfor programmet er endret.
- Feilretting: Fjernet *none member* og gjort *member outline* tydeligere.
- Litt amputert uke pga. en kort vinterferie. Jobbes inn i igjen over de neste ukene.

### 7. uke - Tooltips, hardware-problem, re-installering, feil/nye implem.

Møte m/veileder, 4. mars:

- *Tooltips*: Skulle ifølge planen vært ferdige med dem nå, men vi har tatt for lett på oppgaven for det er veldig mange som må endres. Sender forslag til Kolbein fortløpende slik at han kan komme med innspill.
- Skifter fokus fra feilretting/endringer til å få gått gjennom *tooltips* denne uka.
- Eksentrisiteter: Kolbein har påvist at resultatdiagrammer for modeller med eksentrisiteter blir feil. Bør rettes snart fordi det er viktig å finne ut om dette er en feil i beregningene eller i *GUI*.
- Forslag til nye implementasjoner:



- Vise punktene som har maksimal forskyvning og krefter i resultatdiagrammene. Kolbein synes det er en god idé så dette vil innføres.
- Diskuterte en del mindre endringer i programmet.

Arbeid:

- Feilretting:
  - Registrerte at initaltøyning ikke blir overført til nye *member* ved deling av member. Symbol for tøyning blir borte ved bruk av *undo/redo*.
  - *Arch members*: Ferdig med å rette delingen i *submembers*.
- Hardware-trøbbel på min PC. Brukte mye tid på feilsøking og endte opp med å installere Windows 8. Fikk ikke fikset alle feil med det, men det hjalp.
- Installerte Visual Studio 2013 i samme slengen, og oppdaterte programvaren som brukes for versjonskontroll.

## 8. uke - *User's manual, validering av input, tooltips*

Møte m/veileder, 11. mars:

- Varslet om at funksjonen *X-dof release* er ikke dokumentert i *User's manual*.
- Gikk gjennom forslag til nye *tooltips*. Ble enige om å skrive hvilke hurtigtaster som kan brukes for *controls* i *tooltips*, for eksempel CTRL-O for å åpne filer, etter mitt forslag.
- Ble enige om å legge til *settings*-knappen i *quick access toolbar*, etter mitt forslag. Ble også enige om å endre layout i *application menu* så alt ser mer enhetlig ut.
- Tok opp tråden fra møtet 11. februar: Bestemte oss for å vise en dialogboks dersom man kjører analyse uten å ha lagt til laster. I dialogen vil man da få mulighet til å legge til last (e.g. egenvekt) før analysering.

Arbeid:

- Mye software-trøbbel denne uka. Har jobbet med å finne ut av *Microsoft Office Access Database Engine*. Ulike PC-konfigurasjoner trenger nemlig ulik versjon av dette tilleggsprogrammet, enten 2007 eller 2010. Har lagt til begge i mappen med installasjonsfiler.
- *Tooltips*. *Modelling ribbon page*, *modelling toolbox*, *status bar*, *application menu* og *quick access toolbar* ferdig gjennomgått.
- Feilretting: Validering av *arch members*.

## 9. uke - *Eksport/import, default load combination, feilretting*

Møte m/veileder, 18. mars:

- Diskuterte hvorfor høyre musetast brukes i resultatvisninger, virker rart for meg.
- Arne Aalberg rapporterte at *buckling analysis* ikke fungerer optimalt fordi brukeren varsles på en uforståelig måte når det ikke er lagt til laster i *load combination* før analysekjøring. Generelt ble vi enige om at dersom det er satt default-verdier for *load cases* og *load combinations* så skal brukeren bli informert om det. Hvis ikke skal det ikke settes default-verdier.
- Eksport/import: Gikk gjennom formatet som vi skal bruke i eksport/import-filer. *Joints*, *members*, *cross sections* og *results* skal skrives ut. Bare *joints* skal importeres.
- Diverse diskusjoner om løsningene for *arch members*.

Arbeid:

- Feilretting:
  - *Arch members*: Validering av geometri måtte endres noe igjen.
  - Begynte å rette feilaktig *snap* ved museklikk.

## 10. uke - *Feilretting, release notes, steel design, betatesting*

Møte m/veileder, 25. mars:

- Gikk gjennom bruken av *steel design* i ulike analyser. Vi ble ikke helt enige om hvor man skal bruke det, Kolbein må høre litt med Arne Aalberg og vi må lese i Fredrik Larsens masteroppgave. Må komme fram i *User's manual* at *steel design* er lite testet.

- Jeg påpekte at visualiseringen av *steel design*-utnyttelse må forbedres. Den er rett og slett feilaktig implementert. Fargene som skal vise utnyttelsen interpoleres og det er ikke ønskelig (90 % utnyttelse vil se ut som 100 % fordi fargene blir veldig like).
- Kolbein og Espen vil gjeninnføre visning av musepeker-koordinater i *status bar*. Jeg er enig.

Arbeid:

- Fant ut at jeg har brukt for mye tid på å jobbe med *arch members*. Enkle småfeil skal prioriteres for å få unna noen punkter på listen over feil.
- Feilretting:
  - Lukking av dialogboks for initialtøyning ble rettet.
  - Fjernet/deaktiverte enkelte resultatvisninger og *steel design* for analyser som ikke bør benytte disse funksjonene (dynamiske analyser).
- Jobbet mye med dokumentasjon i form av *release notes*.
- Innførte *DevExpress 13.2.7* og *13.2.8*-oppdatering i kildekoden. Førte ikke til spesielle endringer.
- Betatesting: Laget evalueringsskjema for betatestingen og la det ut. Fikk inn svar ganske fort.
- Leverte *release notes* til Kolbein slik at han kan se alle endringer som er gjort i den nye versjonen Espen har laget (**3.3.0 Developer**). Tilbakemeldingen fra Kolbein var god.

## 11. uke - Tooltips, rapport, validering/formatering, steel design

Møte m/veileder, 1. april:

- Gikk gjennom evalueringene som vi har fått i forbindelse med betatesting:
  - Alle 17 som har svart har brukt **fap2D** i øvingene sine - supert!
  - Fått gode tilbakemeldinger på hva som bør endres og ikke. Vi tar hensyn til dette og Kolbein er enig i at det er en god idé.
- Kolbein kom med tilbakemelding på *release notes*:
  - Overlappende *joints* bør være mulig i programmet for å få en slags 3D-effekt som kan være nyttig i spesielle tilfeller. Bør heller få valget mellom å slette overlappende *joints* eller beholde dem. Jeg skal implementere endringene.
  - Feilaktig *snap* ved museklikk: Jeg har brukt *world coordinates* som kriterie for når et museklikk skal regnes som et flytt. Burde brukt *screen coordinates* slik at det passer bra for alle zoom-nivåer.
- Element-retning: Det er noe som skurrer med elementene i programmet, de skal være definert f.v. mot høyre, men blir det ikke fordi elementretningen er avhengig av retningen man tegnet opp en *member* på. Derfor bør elementretningen redefineres før analyse. Har påvirkning på visning av elementresultater. Espen tar på seg jobben.

Arbeid:

- *Tooltips: Analysis ribbon page*.
- Gått gjennom gamle møtelogger og logger for å summere opp hva som er gjort og hva som må gjøres fremover.
- Jobbet mye med rapportformatering i *LaTeX*.
- Formatering og validering av input i diverse dialogbokser.
- *Steel design*: Startet å endre visualiseringen av resultater.
- Ryddet i *OpenGL*-tegne-metoder for å gjøre dem mer ryddige.

## 12. uke - Steel design, diverse feilretting, settings, selection/markering

Møte m/veileder - Veileder borte på ferie, ikke møte denne uka

Arbeid:

- Jobbet mye med *steel design* denne uka.
- Skalering av resultater, visning av maksimalverdier implementert.

- *Selection*/markering: Jeg har ordnet slik at man nå kan markere (*sub*)*members* ved å trykke innenfor deres *member outline*. Fungerer ypperlig, med mindre man zoomer veldig langt ut.
- Fjernet ubrukte og unødvendige *settings*. Veldig mye rot som må ryddes opp i *settings* generelt. Dokumentasjonen fra tidligere studenter stemmer ikke overens med det vi finner i kildekoden.

### 13. - Påskeferie, litt jobbing med rapport og ny implementasjon

Møte m/veileder - Påskeferie, ikke møte denne uka

Arbeid:

- Jobbet med visning av maksimalverdier i resultatdiagrammer. En liten lilla prikk viser maksimalpunktet.
- Jobbet mye med rapportformatet i LaTeX. Godt å være litt på forskudd slik at det er enklere når skriveingen begynner for alvor i mai.

### 14. uke - Settings, ny implementasjon

Møte m/veileder - Oppstart igjen etter påskeferie, nok å jobbe med og ikke behov for møte

Arbeid:

- Viste seg at endringene jeg gjorde i forrige uke ikke fungerte i alle tilfeller ved visning av diagrammer. Har jobbet med å forbedre dette.
- Har rensket opp en del i *settings*. Måtte bruke en del tid på å forstå hvordan innstillinger lagres og opprettes. Dessuten er dialogboksen med *settings* litt komplisert.

### 15. uke - Eksport/import, formatering/valid., User's manual, steel design

Møte m/veileder, 29. april:

- Omfattende møte, siden det er lenge siden vi har møttes.
- Har gått nøye gjennom eksport/import-format slik at vi kan implementere det de neste ukene.
- Kolbein har funnet feilen som han rapporterte om i forbindelse med stegvis pålasting i *nonlinear analysis*. Vi trenger derfor ikke lete mer etter denne. Ble rettet av Kolbein.
- *User's manual*: Espen og jeg må lese gjennom de første 20 sidene og se om innholdet som Kolbein har revidert stemmer overens med programmet.
- De nye musepeker-koordinatene vises bare i millimeter, men bør også kunne vises i meter. Espen fikser dette.
- *Steel design*: Blir enige om at fargene i *steel design*-utnyttelsesdiagrammet ikke skal kunne endres i *settings*. Dermed kan vi lage en fast fargeskala som blir bedre. Vise knallrød farge for utnyttelse over 100 % slik som før. Jeg ordner dette.

Arbeid:

- Har jobbet sammen med Espen om eksport/import i dag. Samarbeidet har fungert veldig bra og vi har virkelig fått utnyttet versjonskontrollen til fulle denne uka. Eksport av resultater tok lang tid, vi måtte finne ut hvordan man skulle velge ut de *members* som resultatene skulle hentes fra. Vurderte å lage en *user control*, men vi har ikke prøvd dette før så jeg implementerte heller en enkel tekstboks der man kan velge ut *members*.
- Ved å jobbe med eksport-formatet oppdaget vi nye måter å formatere på, som vi benyttet i formateringen av input-felter generelt i programmet. Førte til bedre kontroll over input og bedre visning av output.

### 16. uke - Eksport/import, User's manual, tooltips, settings, selection

Møte m/veileder, 6. mai:

- Eksport/import: Gikk gjennom implementasjonene som er gjort så langt. Diskuterte problematikken med bruk av desimalskilletegn og skilletegn mellom kolonner. Det brukes ulike formater så vi bestemte oss for å la brukeren velge hvilke symboler man vil bruke. Mer fleksibelt, selv om det er mer tungvint.

- *User's manual*: Har gått gjennom endringer. Nummereringen av avsnittene er på plass og det er bra slik at vi kan referere til dem i *tooltips*. Delene om *steel design* må derimot endres i hht. til at jeg har endret visualiseringen.
- Rotasjonsfjærer må prioriteres fordi Arne Aalberg og Petter H. Holmstrøm har opplevd feil med dem.

Arbeid:

- Mer eksport/import.
- Tok opp igjen tråden fra møtet 1. april og gjort endringer i:
  - Overlappende *joints*: Mulighet for å beholde *joints* er lagt til.
  - *Snap* ved museklikk: Har endret definisjonen av museklikk slik at alle flytt av musen nå kontrolleres mot et minimumskriterie i skjermkoordinater (*screen coordinates*).  
Fungerer bedre for alle zoom-nivåer nå.
- *Tooltips*: Gått gjennom alle eksisterende *tooltips* i *analysis* og *results ribbon pages*. Omfattende arbeid, men nå ser alt bedre ut. Referanser til avsnitter i *User's manual* lagt til. Skaper mer enhetlig programopplevelse.
- Tok en statusoppdatering der jeg sjekket loggen o.l. etter punkter som ikke har blitt rettet, men burde vært gjort.
- Mye diverse småfeil har blitt sett på, blant annet ytterligere opprydding i *settings*.
- *Selection* av elementer: Brukt samme tankegang som for *submembers* (se uke 12 for detaljer). Markering av elementer fungerer nå mer generelt. Koden er ryddigere, enklere og kjappere.

## 17. uke - Feilretting, default load combination, nye settings, steel design, markering/selection, betatesting - oppsummering

Møte m/veileder, 13. mai:

- *User's manual*: Vi diskuterer hva som bør legges til i *User's manual* før rapportinnlevering: Eksport/import, *steel design* og *settings*.
- Kolbein vil at man skal *zoom to model* etter at en analyse er ferdig slik at resultatdiagrammer vises fint etterpå. Vi innfører dette, med en *settings*, hvor man kan velge om man vil zoome eller ikke. Jeg mener det er nyttig å kunne velge.

Arbeid:

- Jobbet mye denne uken! Det nærmer seg slutten på programmeringen.
- Overlappende *joints* er nå ferdig implementert. Kolbein er fornøyd også.
- Tok opp tråden fra møtet 18. mars og har nå sørget for at dersom det ikke legges på laster så skal ikke egenvekt legges til i beregningsmodellen automatisk. Brukeren blir varslet i en dialogboks dersom ingen laster er satt på.
- *Settings*: Lagt til en del nye, blant annet for zooming og editering av gravitasjonsakselerasjonen, *g*.
- *Steel design component control*: Gjeninnført bruken av *component control*. Denne har på et merkelig vis blitt fjernet fra koden, uten at jeg finner dokumentasjon på hvorfor.
- *Selection*/markering; av utbøyd forskyvningselement er innført. Forbedret markeringen av *arch members* ved å la brukeren bestemme (i *settings*) hvor presist/nær midtlinjen i *member*-en man må trykke for å få valgt den. Enklere for brukeren.
- Oppsummert betatestingen og ført inn et vedlegg om dette i rapporten. Viktig å poengtere at betatestingen er vektlagt i masteroppgaven.
- Ført inn et vedlegg med figurer av alle endrede *tooltips*. Letter for nye utviklere å holde oversikt over dem nå.

## 18. uke - Rapport (Release notes), dynamiske analyser, nye settings

Møte m/veileder, 20. mai:

- Skrive ut loggfil: Espen og Kolbein har sett på om man kan få skrevet ut loggfilen til brukerens mappe på PC-er i stedet for programfiler-mappen. Begrunnet med at det ikke er skrivetilgang til alle brukere i programfilene (se betatesting uke 3 for detaljer). Lar seg

nok ikke gjennomføre på kort sikt, for da må en del endres i beregningskjernen, for det er den som skriver ut logg.

- Kolbein informerer om at rapportformatet er B5 og det vil si at vi må justere opp skriftstørrelsen og figurene våre. Rapporten vil bli en del flere sider!

Arbeid:

- Rapportarbeid: Skrivning av *release notes* og vedlegg. Jobbet mye med disposisjon og struktur på rapporten forrige uke og denne uka. Begynner å komme på plass. Lagt opp strukturen i hovedkapittelet "Endringer i **fap2D**".
- Nye settings: Advarsler for *arch member*-geometri og korte elementer kan fjernes fra programmet ved å huke av for dette i *settings*.
- *Release notes*: Bestemte oss for å samle alle *release notes* i ett dokument så det blir mer lesbart og enklere å følge med for leseren i rapporten vår. Vi har jo laget mange interne *release* (nye versjoner av **fap2D**), men det kan bli vanskelig å følge med. Likevel brukes *release*-numrene for å vise at det er gjort endringer i implementasjonr undervegs ettersom vi har testet programmet.
- Dynamiske analyser: Jeg har prøvd å gjennomføre noen analyser og kontrollere resultater. Har brukt øvingsoppgaver fra mekanikk-fagene mine ved NTNU. Rekker ikke omfattende testing dessverre. Finner ikke gode testeksempler for tvungne dynamiske analyser og jordskjelv, det er ikke gjort i en håndvending.

## 19. uke - Rapportskrivning, siste endringer i koden

Møte m/veileder, 27. mai:

- Diskuterte disposisjonen/rapporten og Kolbein synes den ser bra ut.
- Espen og jeg vil ha inn alle enheter i *settings* slik at man kan velge dimensjonene som brukes. Grunnen er at det vil bli enklere å modellere veldig små og store tallverdier på denne måten. Kolbein er enig.
- Siste kontroll av hvor *steel design*, *resultant forces* og *reaction forces* skal vises. Fortsetter omtrent som planlagt - *steel design* kun for lineær og ikke-lineær statistisk analyse.
- *Steel design*: Ble enige om å vise varseldialoger dersom brukeren prøver å kjøre analyse på konstruksjoner med tverrsnitt og *member type* som ikke kan brukes.
- Informerte Kolbein om unøyaktig utregning av elementplassering langs *arch members*. Vi skal prøve å rette feilen, men den kan være i beregningskjernen.

Arbeid:

- Rapportarbeid. Jeg har skrevet inn mye i de ulike kapitlene i rapporten, mens Espen har jobbet godt med *release notes* og hovedkapittelet "Endringer i **fap2D**".
- Innførte alle resterende *settings* og gjorde de andre endringene vi ble enige om på møtet. Gikk ganske raskt så rapporten led ingen nød.
- Fant ut av feilen med elementplassering skyldtes forveksling mellom elementdefinisjon langs *arch* (bue) og mellom noder (rettlinjet). Ble rettet greit.

## 20. uke - Siste finpuss på rapporten

Møte m/veileder, 3. juni:

- Innleveringsprosedyrer.
- Gjennomgang av utkast til rapport.
- Generere endelig versjon av **fap2D**:
  - En med lisens ut september 2014
  - En uten lisenskontroll.
  - En uten lisenskontroll, som ikke skriver ut loggfil slik at **fap2D** kan brukes på datasalene til instituttet. Dermed blir det ingen problemer med skriverrettigheter.

Arbeid:

- Siste finpuss på rapporten denne uka. Ble litt travelt, men gikk greit. Synd at vi ikke fikk tid til mer testing av analyser, men vi må innse at vi ikke har kapasitet til det.

## D.2 Logg - Espen Skogsrud

### Logg - Espen Skogsrud

Loggen viser progresjonen i arbeidet med masteroppgaven, ved å vise hva som har blitt gjort hver dag. Loggen er i hovedsak ment for eget bruk. Derfor presenteres bare ukentlig oppsummering fra loggen i dette dokumentet. Detaljer om endringer i kildekoden presenteres ikke her, bare kommentarer til endringene i hht. punktene i *Release notes*.

#### Før offisiell start (Kalenderuke 52 - 2) - Oppstart

Møte m/veileder, 8. januar:

- Diskuterte prosjektoppgaven og videre arbeid med den som utgangspunkt
- Bestemte problemer som skal fokuseres på (diagrammer og *arch members*)
- Diskuterte organiseringen av arbeidet

Arbeid:

- Leste vurdering av prosjektoppgaven

#### 1. uke (Kalenderuke 3)

Møte m/veileder, 14. januar:

- Planlegging av tester - enkle tester som sammenligner med Focus
- Diskuterte betatesting av **fap2D** i KMEK
- Diskuterte prosjektoppgavene

Arbeid:

- Begynte å se på N- og V-diagrammer og ulike løsninger - noe implementasjon ble gjort
- Ble enige om milepæler for prosjektet, en overordnet oversikt over oppgaven og prioriterte oppgaver og nødvendig tid til hver oppgave.
- Vurderte og undersøkte muligheten for å gå over til ny måte å rapportere bugs

#### 2. uke (Kalenderuke 4)

Møte m/veileder, 21. januar:

- Ble enig om endelig løsning av N- og V-diagrammer
- Diskuterte inndeling, størrelsesbegrensning og modellering av *arch members*

Arbeid:

- Jobbet med og implementerte ferdig N- og V-diagrammer
- Undersøkte feil om opplager og joints som ikke ble tegnet

### 3. uke (Kalenderuke 5)

Arbeid:

- Rettet arch-members slik at de kan snappes til en annen member samtidig som den har maks utstrekning.
- Undersøkte feil ved tegning av arch-members ved klikk og mouseHover.
- Rettet feil ved tegning av for kort bjelke.
- Svarte på og loggførte bugs sendt inn av brukere. Fant og løste bug for tegning av joints og opplager i resultatfanen.

### 4. uke (Kalenderuke 6)

Møte m/veileder, 04. februar:

- Vurdering av problemer med fap2D på sal
- Diskusjon om minste elementstørrelse og advarsler
- Feilmeldinger og versjoner ble også diskutert

Arbeid:

- Undersøkte og dokumenterte lagring og lasting i fap2D
- Vurderte løsninger av elementlengdebegrensninger. Implementerte det som metoder i for members og submembers. Gjør sjekk i member properties, mesh properties og meshing.

### 5. uke (Kalenderuke 7)

Møte m/veileder, 11. februar:

- Diskuterte import/eksport funksjonalitet og kompleksitet
- Ble enige om å fjerne ghost members
- Elementstørrelsen ble valgt til å baseres på minste tverrsnitt for å være konservativt

Arbeid:

- Laget ny versjon (3.2.0)
- Justerte advarsler for elementlengde for å ikke dukke opp der elementstørrelsen er uendret.
- Refaktoriserte en del kode i member properties i sammenheng med elementlengde
- prøvde å skrive noen enhetstester for joints - litt usikker hvordan jeg skal teste

### 6. uke (Kalenderuke 8)

Møte m/veileder, 19. februar:

- Diskuterte navigering og valg av elementresultater
- Prøvde å finne løsning på arch member inndeling

Arbeid:

- Diskuterte og implementerte endringer av lagring og lasting. Modeller kan nå lagre og laste selv om det er mangler. Gjorde flere mindre justeringer av dialoger, og dokumenterte endringer.
- La inn og vurderte en del rapporterte bugs

### 7. uke (Kalenderuke 9)

Møte m/veileder, 25. februar:

- Diskuterte arch member inndeling
- Vurdering av mulighet for endring av enheter. Det virker greit implementert, men hvis det er glemt et sted kan det gi alvorlige feil.

Arbeid:

- Rettet omregning av rotasjonsstivhet.
- Lagt til fortegn på aksialkraft
- Diskuterte hvordan visning maks respons i diagrammet kunne implementeres
- Jobbet med exceptions for arch members. Noe med Distance() og CurveDefine().
- Rettet en feil der tooltip prøvde å finne et objekt som ikke eksisterer

### 8. uke (Kalenderuke 10)

Møte m/veileder, 4. mars:

- Blitt enig om knapper for elementnavigering. Markeringen av elementer må få maksstørrelse
- Diskusjon av problemer med eksentrisiteter
- Bestemte at spenningsdiagrammene kan være ganske enkle. Trenger kun vise relative maks lengder med farger.

Arbeid:

- Jobbet med og dokumenterte forskyvning av forskyvning som vektorlengde.
- Vurderte løsninger om navigering av elementresultater. Implementerte løsningen.
- Undersøkte hvorfor circular arch members med full utstrekning ikke kunne markeres. Noe med at vinkelen vinkelen ved et joint er double.NaN
- Diskuterte gjennomgang av dialogbokser

### 9. uke (Kalenderuke 11)

Møte m/veileder, 11. mars:

- Diskuterte feil for \0 returnert fra FindShortestDistanceToCurve()
- Ble enig om format på import og eksport
- spenningsdiagrammer er kun lineære, men kan ha ulik avstand til nøytralakse

Arbeid:

- Fikset en feil der minste elementlengde krasjet
- Debugget feil for FindShortestDistanceToCurve() der det ble returnert NaN.
- Undersøker feil i Mouse Events for modellering av members.
- Refaktoriserte og endret UC\_stresses for å vise spenninger bedre.
- Jobbet litt mer med refaktorisering av member properties. Mange kombinasjoner av members gjør det litt komplisert.



**10. uke (Kalenderuke 12)**

Møte m/veileder, 18. mars:

- Bekreftet at bar members kun har konstant tverrsnitt
- Resultants skal ikke vises i dynamiske analyser
- Vurdering av markering med høyre musetast. Høre på betatester. Ellers, la den være.
- Ble enige om flere detaljer om import/eksport
- Trolig gjøre noe testing av dynamiske analyser fra Chopra
- Vil prøve å gjeninnføre negativ radius

Arbeid:

- Jobber mer med member properties. Noen tverrsnittsregler er ikke konsekvente. Legger inn felter for <multiple> og <interpolated>. Noe vanskelig å jobbe med dropdown lists med bilder.
- Dokumenterte tidligere arbeid. F.eks. vektorlengder og navigering.
- Funnet noe feil om resultanter.

**11. uke (Kalenderuke 13)**

Møte m/veileder, 25. mars:

- Vurderte lisensløsninger. Kanskje IDI har en løsning, trolig noe med VPN.
- Det ble bestemt at egenslast ikke skal settes på automatisk. Bruker må selv inkludere.
- Nye feil om beregning med flere modeller åpne, og om ikkelineære beregninger undersøkes.

Arbeid:

- Jobber med feil i modellering av eksentrisiteter. To eksentrisiteter samtidig skaper problemer. To definisjoner av eksentrisiteter som var håndtert ulikt flere steder.
- Jobbet med en "How To" for å lage installasjonsfiler.
- Laget ny versjon (3.3.0)

**12. uke (Kalenderuke 14)**

Møte m/veileder, 1. april:

- overlappende eller veldig nære joints skal vises rødt, og man burde få valg om en bedre visning i en dialogboks. Må vurdere hva som er minste avstand
- Ny feil om at modellen "snapper" når den klikkes på ble diskutert
- Diagrammer for elementer vises opp-ned. Bør endre alle members slik at alle ligger i en logisk orientering. Må også da gjeninnføre arch members med negativ radius.
- Overlappende joints skal ikke slås sammen hvis de er flyttet over hverandre. man skal få et valg

Arbeid:

- Mer om eksentrisiteter. Nesten ingen steder blir det tatt hensyn til at joints og enden av members (eller noder og enden av elementer) kan være adskilt av eksentrisiteter.
- Dokumenterte om eksentrisiteter
- Undersøker hvorfor assosierte filer ikke åpnes direkte i programmet. Trykker man på en .fap2D fil skal den vises med én gang
- Undersøker feil der feil modell blir regnet på
- Et par mindre endringer i member properties

### 13. uke (Kalenderuke 15)

Arbeid:

- Bestemte oss for organisering av rapport og hva vi skal ta med og utelate fra rapporten
- implementerte visning av current mouse coordinates
- Litt refaktorisering av member properties. Noe mer arbeid enn jeg trodde. Burde kanskje endre pattern brukt i dialogen.

### 14. uke (Kalenderuke 16)

Påskeferie

### 15. uke (Kalenderuke 17)

Arbeid:

- Undersøker om hvorfor negativ radius/høyde ikke fungerer. Virker som vi har fikset de fleste feilene.
- Endret en del valideringsmetoder for å ta hensyn til negativ radius
- Tenkte ut mulighet for manuell justering av arch radius/høyde
- Fjernet mulighet for varierende tverrsnitt langs bar, cable og strut members

### 16. uke (Kalenderuke 18)

Møte m/veileder, 29. april:

- ble enige om funksjon og utseendet av koordinatfelt for musepeker
- Diskuterte detaljer rundt en eksportfunksjonalitet. skilletegn og antall signifikante siffer trenger ekstra oppmerksomhet. Eksponentform bør kunne brukes.
- Konvertering av memberretning skal skje rett før meshing for at bruker ikke skal tenke på det.

Arbeid:

- Debugger feil MKey verdi for ikke-lineær. Kun foreskrevet forskyvning gir problemer.
- Opprettet dialogboks og knapper for eksport-funksjonalitet. Vurdert hva som skal være valgbart og hvordan output skal se ut.
- Endret koordinatfelt for peker til å reflektere enhetene i resten av programmet
- La til navigeringsknapper i flere dialogbokser der de burde være med
- Opprettet dialogboks for import

**17. uke (Kalenderuke 19)**

Møte m/veileder, 6. mai:

- Gikk gjennom eksport- og importdialogene frem til nå. Fungerer greit. Må samkjøres.
- Det ble foreslått et notat om kjente problemer

Arbeid:

- Jobbet videre på Import for å gjøre lesing av rader lettere. Hensyn må tas for desimaltegn, id og start-/sluttrad. Lager en "smart"-parser som prøver seg fram automatisk
- Ny setupfil (3.3.1)
- Vurderer om members burde roteres for å vise bedre resultater.
- Gikk gjennom endringer i UM
- Vurderte plassering av settingsfiler og lignende. Tror Appdata-mappen er best. Kanskje Log-filen kan legges dit også?
- Endret utseendet på dialog for response arameter og fjær ved hjelp av layoutcontrol.
- Fikset oppdatering av y-akse i ikke-lineær fjær

**18. uke (Kalenderuke 20)**

Møte m/veileder, 13. mai:

- Diskuterte enheter. Alt i Frame2D er i SI, fap2D bruker enheter etter tidligere dokumentasjon, men noen flere kan legges til.
- Versjonsnummer. Ikke helt 4.0.0. Kanskje 3.1 siden det er en stabil versjon.
- Kanskje FORTRAN kan skrive til appdata. Gikk gjennom noen endringer i UM.

Arbeid:

- La til mulighet for å endre enheter for rotasjonsstivhet og frekvens
- Laget ny versjon (3.5.0)
- Fikset feil der spenning ble feil utregnet. Enheter ble endret for tidlig.
- Endret opptegning av eksentrisiteter i dialogboks
- Implementerte validering av arch members endret av eksentrisiteter
- Fikset feil der det er flere MoveJoints, eller MoveMember og moveJoint. Movejoint må endres litt den kan endre et joint flere ganger, i motsetning til å bare endre én member for moveMember. Endret også nullsetting av dem.
- Liten fiks der infotekst krasjer programmet
- Testet at member snus korrekt. Liten endring for tverrsnitt i submembers

**19. uke (Kalenderuke 21)**

Møte m/veileder, 20. mai:

- Appdata er grei mappe å skrive til. Print flagg for logg bør være satt til 1. Bør ikke settes til noe annet av bruker.
- Diskuterte dokumentering av endringer i rapporten. Fordele i release notes med detaljer, og oppsummert i hoveddelen. Kun det viktigste skal i hoveddelen, referanser der interessant.
- Bestemt at bruker kan deaktivere advarsel om mesh.

Arbeid:

- Ryddet opp i dokumentasjonen for gamle endringer fra den siste måneden.
- Sletter nå resultatene når noe endret slik at de ikke kan feilaktig vises. Er dette litt for voldsomt? Kan ikke få de tilbake med undo ...
- Laget ny versjon (3.5.1)
- Endret filstien til midlertidige filer
- Flytter litt rundt på instillinger i settings
- Litt ekstra testing av eksentrisiteter
- Skrev de noen release notes inn i LaTeX
- Gjennomgang av innhold i rapporten. flytter litt rundt på oversikt over hva som er endret.
- fikset feilen der sletting av overlappende members slettet joints flere ganger

### 20. uke (Kalenderuke 22)

Møte m/veileder, 27. mai:

- Litt diskusjon om oppsettet av oppgaven.
- Klargjort om årsak og begrunnelse for elementinndelingsadvarsel
- Prøvde å finne feilen om elementavstand langs arch members.

Arbeid:

- Skrev resten av release notes inn i latex
- Ny versjon (3.5.2)
- Jobbet med rapporten. Skrevet alle endringer inn i samlet tekst. Trenger fremdeles en del bilder og referanser.
- Fikset feil om nodeplassering langs arch members

### 21. uke (Kalenderuke 23)

Møte m/veileder, 3. juni:

- Byttet elementinndelingskrav til 5% istedenfor 10%. Alle analyser bruker diagonal decay test med  $\epsilon = 10E-08$
- Diskusjon om rapporten. Bildebruk, språk etc.

Arbeid:

- Jobbet med rapporten

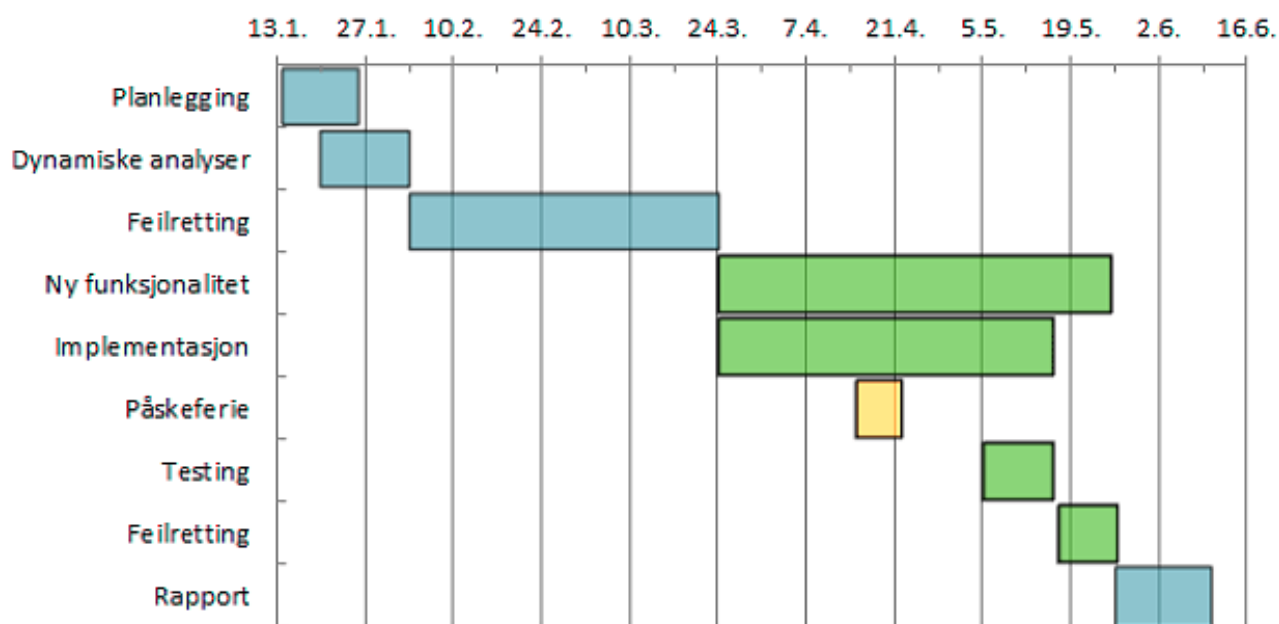
### 22. uke (Kalenderuke 24)

Arbeid:

- Ferdigstilling og innlevering

# Tillegg E

## Gantt-diagram



Figur E.1: Tidsplan for masteroppgaven.

# Tillegg F

## Programvare

### F.1 Systemkrav for fap2D

Systemkravene for **fap2D** er hentet fra kapittel 2 i **fap2D** sin *User's Manual* [7] og fra [8].

#### F.1.1 Krav til hardware

Tabell F.1: Minimumskrav til hardware.

Prosesor	1.5 GHz
Minne	2 GB
Lagringsplass	80 MB
Skjermopløsning	1366 x 768
Grafikkort	Grafikkort med støtte for <i>OpenGL 2.1</i>

Det anbefales å ha nyeste driver til grafikkortet installert. De nyeste driverne kan installeres fra nettsiden til leverandøren av grafikkortet. Slik finner man ut hvilket grafikkort en PC har:

1. Høyreklikk på “Min datamaskin” og gå inn på “Egenskaper”.
2. Klikk videre inn på “Enhetsbehandling” som ligger i en liste til venstre under egenskapene. Under “Enhetsbehandling” finner man leverandøren av grafikkortet.
3. Besøk nettsiden til leverandøren og følg instruksene der for å laste ned nyeste driver.

Tabell F.2: Leverandører og deres hjemmeside for grafikk-driverne (per 1. juni 2014).

Intel	<a href="http://www.intel.com/p/en_US/support/detect/graphics">http://www.intel.com/p/en_US/support/detect/graphics</a>
Nvidia	<a href="http://www.nvidia.com/Download/index.aspx?lang=en-us">http://www.nvidia.com/Download/index.aspx?lang=en-us</a>
AMD	<a href="http://support.amd.com/us/gpudownload/Pages/index.aspx">http://support.amd.com/us/gpudownload/Pages/index.aspx</a>

## F.1.2 Krav til software

Tabell E.3: Krav til software.

Støttede operativsystemer	Tilleggsprogrammer
Microsoft Windows Vista, 7, 8 og 8.1	Microsoft .NET 4.5
Eldre utgaver som Windows XP støttes ikke	Microsoft Office Access Database Engine

Dersom Microsoft .NET 4.5 ikke er installert kan det enkelt installeres via en fil som er inkludert i en egen mappe i installasjonen til **fap2D**. Microsoft Office Access Database Engine følger med Microsoft Office. Dersom Microsoft Office ikke er installert kan Access Database Engine installeres via filer som er inkludert i en egen mappe i installasjonen til **fap2D**.

**OBS:** Ulike PC-konfigurasjoner krever ulik versjon av Access Database Engine. 2007- og 2010-versjon finnes i installasjonsmappen. Dersom den ene ikke fungerer, prøv den andre.

## F.2 Programvare brukt i utviklingen

Følgende programmer har blitt installert i forbindelse med utviklingsarbeidet i **fap2D**:

Tabell F.4: Programvare for utvikling.

Programvare	
Utviklingsplattform	Microsoft Visual Studio Professional 2012 Microsoft Visual Studio Ultimate 2013
Programmeringsteknologi	C# / Microsoft .NET 4.5
Grafikk-rammeverk	Tao Framework 2.1.0
GUI/design-rammeverk	DevExpress 13.2.8 - WinForms
SVN-klient	TortoiseSVN 1.8.5
Plugin for SVN	Visual SVN 4.0.5
Innlogging på SVN server	PuTTY 0.63

Visual Studio er brukt som utviklingsmiljø. Programmeringsspråket som er brukt er *C#*, i kombinasjon med programmeringsplattformen Microsoft .NET. Tao Framework er et *C#*-bibliotek som gir tilgang til grafikkbiblioteket *OpenGL*. Installasjonsfilen for Tao finnes i en egen mappe i installasjonen til **fap2D**. DevExpress WinForms er et tillegg som gjør arbeidet med å lage grensesnittet enklere. Kolbein Bell har brukerkonto hos DevExpress. Lisensen gikk ut i mars 2014. Det er foreløpig bestemt at lisensen ikke skal fornyes.

TortoiseSVN er brukt som klient for versjonskontroll (*SVN*). VisualSVN er et tillegg som integrerer TortoiseSVN i Visual Studio. Dermed kan alt utviklingsarbeid gjøres inne i Visual Studio. PuTTY er brukt for automatisk innlogging på serveren hvor kildekoden er lagret, slik at man slipper å oppgi passord hver gang man gjør endringer i kildekoden.

# Tillegg G

## Forklaring av kildekoden

### G.1 Navnekonvensjoner

Tabell G.1 gir en oversikt over navnekonvensjonene som er brukt for variabler, metoder, klasser osv. i kildekoden til **fap2D**. Den er basert på tilsvarende tabell fra kapittel E.1 i [6]. De andre retningslinjene som er fulgt for å skrive samstemt kode finnes også i det kapittelet.

Det blir sporadisk referert til navnekonvensjonene i vedlegg A *Release notes* - Versjonsmerknader.

Tabell G.1: Navnekonvensjoner brukt i kildekoden. [6]

Formatering	Beskrivelse
camelCasing	Brukes for å navngi variabler som ikke er private for én klasse. Flere klasser kan dermed få tilgang til den.
<u>camelCasing</u>	Understrek brukes til å navngi lokale og private klasser og variabler. (Dette gjør det enkelt å skille dem fra andre variabler og bruke hurtigfullføringsfunksjonen til Visual Studio).
PascalCasing	Brukes for alle klasser, typer, metoder og navnerom, samt for navn som består av flere ord.
PrePascalCasing	Et valgfritt prefiks brukes for å indikere at klassen arver fra en klasse i standard-biblioteket. Prefikset angir hvilke klasse det arves fra.
KONSTANT	Brukes for konstante verdier.



## G.2 Prosjektoversikt

Oversikt over prosjektene som kildekoden består av. Hentet fra Skogsruuds prosjektoppgave [11].

### Prosjektoversikt for fap2D

Opplisting og kort beskrivelse av prosjektene i **fap2D** per **våren 2014**.

Klassene er fordelt på 9 *prosjekter* (mappetype i Visual Studio) og ofte delt inn i mapper for delklasser, derfor har hvert prosjekt en beskrivelse, og hver delklasse en separat beskrivelse.

Visual Studio deler programmer inn i *solutions*, med *prosjekter/projects* som undermapper, og *folders/mapper* som undermapper av *projects*.

#### *fap2D.Gui* (prosjekt)

Brukergrensesnittet.

Det største prosjektet med 14 delklasser.

[FrmMainGui](#) er klassen som instansieres først og som starter alle andre prosesser.

Inneholder alle dialogbokser i mappen [FrmDialogs](#).

Instansierer [FrmModel](#) (modellpanelet) der alt av tegning og modellerings-*events* håndteres.

FrmModel holder en instans av [StructuralModel](#) (konstruksjonsmodellen) og

[ComputationalModel](#) (beregningsmodellen og resultater fra den).

Klassene er hovedsaklig fordelt i mapper (se nedenfor), men det er også diverse klasser for bl.a. validering av inndata (til tekst-/tallfelt).



#### FrmDialogs (mappe)

Dialogvinduer for hver fane. Delt inn i **Modeling**, **Loading**, **Analysis** og **Results**. I tillegg er en mappe **Other** med dialogvinduer som gjelder alle faner.



#### FrmMainGui (mappe)

Mappe for hovedklassen til GUI'et. Håndterer GUI-operasjoner relatert til alt utenfor tegnevinduet/modelleringspanelet. Dette inkluderer knapper, *ribbons*, *ribbongroups*, faner etc.

Klassen er delt inn i delklassene:

- FrmMainGui - Den overordnede delklassen
- .AnalysisPage - Kontrollerer analysefanen
- .ApplicationMenu - Kontrollerer applikasjonsmenyen (lage ny eller åpne modell etc.)
- .BottomPanel - Kontrollerer bunnpanelet (feilmeldinger og statusoppdateringer etc.)
- .BoundaryConditions - Kontrollerer knapper for grensebetingelser

- .LeftPanel - Kontrollerer venstre panel (informasjon om *joints* og *members*)
- .LoadingPage - Kontrollerer lastfanen
- .ModellingPage - Kontrollerer modelleringsfanen
- .RecentFiles - Kontrollerer visning i application menu av nylig åpnete modeller
- .ResultsPage - Kontrollerer resultatfanen
- .SettingsPage - Kontrollerer globale innstillinger (ikke implimentert)
- .SkinGallery - Kontrollerer endring av programutseende
- .StatusBar - Kontrollerer statuslinjen (ikke implementert)
- .ToolBox - Kontrollerer verktøypanelet (gridinndeling, zoomnivå etc.)



### FrmModel (mappe)

Mappe for tegnevinduet/modelleringspanelet. Håndterer plassering og redigering av modellobjekter (*joints*, *members*, etc.).

Klassen er delt inn i følgende delklasser:

- FrmModel - Den overordnede delklassen
- .BackGroundWorker - Bakgrunnsarbeider (animasjon av resultater)
- .InfoText - Informasjon om modellobjektet ved markøren (mao. tooltip)
- .OpenGL - Tegning av modell og modellobjekter
- .Popup - Menyknapper ved høyreklikking på modellobjekter



### Mouse (mappe)

Mappe for musepeker og dens utseende i ulike situasjoner.



### Settings (mappe)

Mappe for innstillinger av ulike deler av GUI.

## *fap2D.StructuralData* (prosjekt)



Konstruksjonsmodellen.


Definerer geometrien med *members* og *joints*.

Definerer modellegenskaper som laster, materiale, tverrsnitt, punktmasse, fjær osv.

Prosjektets viktigste oppgave er å håndtere informasjon, derfor gjøres få beregninger her. All visuell presentasjon av konstruksjonsmodellen håndteres av [FrmModel](#).

En instans av hovedklassen `StructuralModel` inneholder all geometrien og egenskapene. Alle modellegenskapene instansieres av klasser i mappene nedenfor.

Klassene `DrawingPointCollection` og `ResponseParameters` inneholder henholdsvis koordinatene til alle konstruksjonsobjektene, og responsparameterne.

 **Lists (mappe)**

Klasser som inneholder alle *joints* og *members*. Disse instansieres av klasser i mappen [StructuralItems](#).

 **Loading (mappe)**

Klasser for å lage laster og lastkombinasjoner i konstruksjonsmodellen. Dette inkluderer punkt og linjelaster, *loadpaths*, foreskrevne forskyvninger, lasttog, influenslinjer og andre.

 **MaterialAndCrossSection (mappe)**

Klasser for standard og egendefinerte materialer, og for standard og egendefinerte tverrsnitt. I sammenheng med egendefinering finnes det klasser som utfører beregninger av egenskaper basert på inndata.

 **StructuralItems (mappe)**

Klasser for ulike fysiske konstruksjonsobjekter som tilsvarer modellobjekter i [FrmModel](#). Dette innebærer blant annet *joints*, *members*, punktmasser, fjærer og eksentrisiteter.

 **StructuralSettings (mappe)**

Klasser som lagrer innstillinger for konstruksjonsmodellen. Det brukes globale standardinnstillinger ved instansiering som senere kan endres av bruker. Dette innebærer blant annet kosmetiske innstillinger som gridstørrelse, visning av symboler og zoom, og standardverdier for demping, egenfrekvenser og antall moder.

## *fap2D.Computations (prosjekt)*

Beregningsmodellen og resultatene av beregningene.

Opprettes direkte fra en instans av `StructuralModel` (i prosjektet [StructuralData](#)). En private metode `ComputationalModel.Analyses` utfører analysen på beregningsmodellen ved hjelp av .ddl rutiner fra **Frame2D**.

En klasse `Frame2D` er en wrapper for FORTRAN-rutinene i **Frame2D**, og inneholder en delklasse som håndterer feilmeldinger fra beregningene.

Det blir brukt noen klasser for å støtte beregningene av lasttog i sanntid. I tillegg er det klasser som instansieres som resultater for noder, elementer og laster.

## *ComputationalModel (Klasse)*

Den viktigste klassen i prosjektet, og som også er delt inn i sju delklasser. Den transformerer en konstruksjonsmodell (`StructuralModel`) til en beregningsmodell som kan benyttes i analyser fra **Frame2D** API'et. Resultatene returneres hit etter en analyse, og disse kan tegnes i resultatpanelet med `ComputationalModel.Drawing`. Resultatpanelet er en erstatning for modelleringspanelet.

Delklassene er:

- ComputationalModel - Hovedklassen som inneholder selve modellen
- .Analyses - Styrer beregningene av modellen
- .Drawing - Tegner resultatene i modellpanelet
- .KeyParameters - Lager viktige parametere for bruk i beregningene
- .LoadTrain - Kontrollerer lasttoget i resultatene
- .ModelTools - Manipulering av beregningsmodellen og dens mesh
- .Tables - Tabellene brukt i beregningene av Frame2D



### SteelDesign (mappe)

Inneholder én klasse, DesignCalculations, som utfører ståldimensjonering. Det kan utføre elastisk kontroll, tverrsnittskontroll, og komponentkontroll. Denne benytter de nasjonale parameterne og utfører alle beregningene selv.

### fap2D.Storing (prosjekt) C#

Lagring og åpning av modellen og innstillinger.

StructuralModel (se [StructuralData](#)) inneholder felt som skal lagres med henholdsvis modellen og innstillingene. Hver av klassene (i StructuralModel) har innebygde åpne- og lagremetoder.

StoringUtilities brukes til å styre åpne- og lagringsprosessen. Benytter seg av *Singleton pattern* og ADO.NET rammeverket. Innebygde parametere (tverrsnitt, jordskjelvdata, materialer og nasjonale parametere) har egne databaseklasser.

Her ligger kode for undo/redo som fungerer ved å lage kopi(er) av StructuralModel ([se GUI](#)).

### fap2D.Utilities (prosjekt) C#

Hjelpeklasser for bruk i hele programmet.

Klasser for å håndtere/regne på blant annet farger, matriser, punkter og krumme *members*.

Inneholder klasser forbundet med OpenGL. MultiSampleGLControl brukes av blant annet FrmModel ([se GUI](#)). Dette er en egenlagd control som OpenGL-metoder kan tegne på, dvs. den kontrollerer grovarbeidet slik at andre klasser enkelt kan utføre selve tegningen.

Andre klasser er for kontroll av kameraet, koordinatpunkter, objektmerking og hjelpeklasser. Noen av disse er sortert i mappene nevnt nedenfor.

### ErrorList (mappe)

Hånterer alle feilmeldinger og advarsler som oppstår vedkjøring av programmet. Disse vises vanligvis i feilmeldingspanelet som er en del av bunnpanelet.

### Exceptions (mappe)

Samler alle *exceptions* som er spesifikke for dette programmet. Det er foreløpig liten nytte av disse.

### GlobalSettings (mappe)

Inneholder globale innstillinger av programmet forbundet med utseende og standardoppførsel. Den består av én klasse med flere delklasser som stor sett inneholder *properties (getters/setters)*:

- GlobalSettings - globalt utseende av programmet
- .ComputationalModel - globale innstillinger for beregningsmodellen
- .Constants - diverse konstanter til beregninger
- .CrossSection - globale innstillinger for tverrsnitt
- .Enums - *enum*'er for globale innstillinger
- .Frame2D - globale innstillinger for **Frame2D**-oppførsel
- .Gui - globale innstillinger for brukergrensesnittet
- .NumericalNoise - globale innstillinger for behandling av numerisk støy
- .OpenGL - globale innstillinger for OpenGL
- .StructuralModel - utseende og oppførsel av konstruksjonsmodellen

### InformationList (mappe)

Informasjonslister i sammenheng med MemoEdit (del av DevExpress) og lasttilfeller.

### OpenGL (mappe)

Verktøy for OpenGL-kameraet og for initialisering, *repainting*, koordinattransformasjoner, zooming etc.

### Units (mappe)

Inneholder klasser for enheter og konvertering mellom enheter bruker ønsker og enheter programmet bruker.

## *fap2D.ModelData* (prosjekt)



Abstrakte klasser brukt av alle andre prosjekter.

Den abstrakte klassen `DrawableObjects` arves av andre klasser i dette prosjektet. Grafnoder og linjer er klasser som arver fra den. Det er også en klasse for den overordnede modellen (trolig global informasjon som identifiserer den spesifikke modellen), og et interface for krumme objekter.

Inneholder klasser for inndeling av koordinatsystemet og annet som er fordelt i mapper:...



### Grid (mappe)

Inndeling av koordinatsystemet (*grid*). Avstand og synlighet av *snap*-punkter. Denne klassen utfører også opptegningen av grid'et.



### ModelSymbol (mappe)

Håndterer symbolene som vises for konstruksjonsmodellen. Holder styr på filnavn til bildefiler av symbolene.



### Templates (mappe)

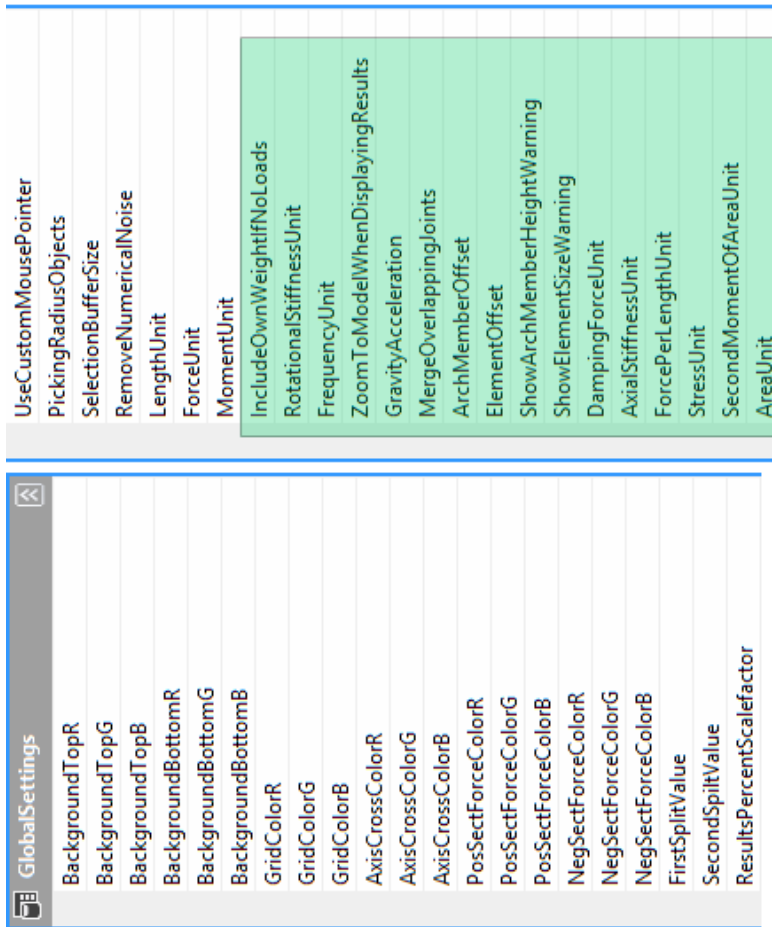
Mal for oppretting av dialogbokser.

## *fap2D.Design* (prosjekt)

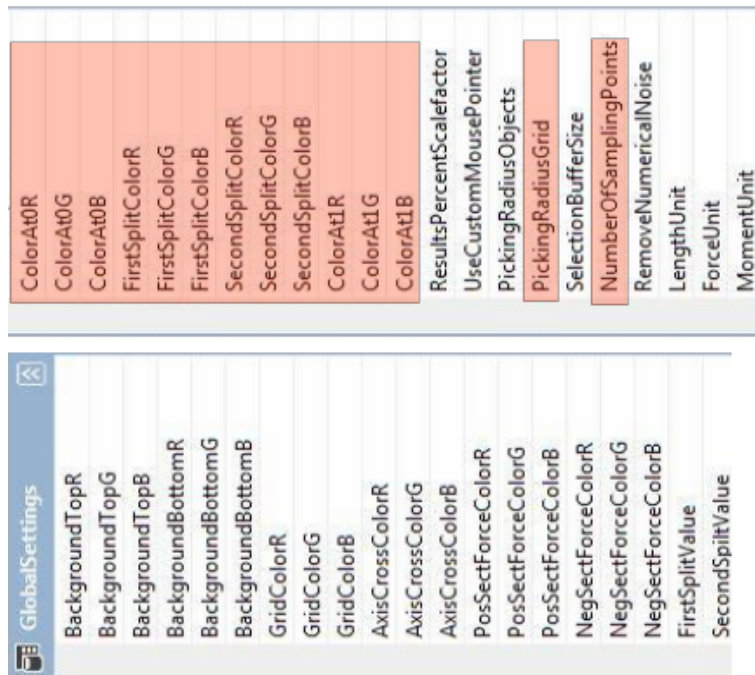
Kapasitetskontroll.

Nasjonale parametere og beregninger relatert til dem.

### G.3 Datasett med *global settings*



Figur G.1: *Global settings* for **fap2D 3.1**. De grønne er nye.



Figur G.2: *Global settings* for **fap2D 3.0**. De røde er fjernet i **3.1**. Hentet fra E.2 i [9].

## G.4 Nøkkeltall - *Code metrics*

*Code metrics* er nøkkeltall som kan hentes ut i Visual Studio. Tallene beskriver en del egenskaper ved kildekoden til et program. De har blitt brukt av tidligere utviklere på **fap2D**, for å se hvordan programmets egenskaper har utviklet seg. Derfor sammenlignes *code metrics* for versjon **3.0** og **3.1** her.

### G.4.1 Forklaring av nøkkeltallene

Forklaringene er hentet fra [14], og fritt oversatt (forenklet) til norsk.

#### **Maintainability index (MI)**

Verdi fra 0 - 100 som beskriver hvor utfordrende det er å vedlikeholde koden. Jo høyere verdi, jo lettere er det å vedlikeholde. Denne bør ikke være under 20.

#### **Cyclomatic complexity (CC)**

Måler kompleksiteten i koden ved å beregne antall ulike stier programmet kan gå gjennom kildekoden på. Dette sier noe om hvor vanskelig det er å kontrollere og ta hensyn til alle de mulige scenariene i programmet.

#### **Depth of inheritance (DI)**

Beregner den lengste stien i klassehierarkiet i kildekoden, fra rotklassene (som ikke kaller noen klasser) og opp til klassene på toppnivå (som ikke blir kalt av noen klasser). Desto lengre sti, desto verre er det å finne ut hva som er det opprinnelige opphavet til en variabel eller metode.

#### **Class coupling (CCo)**

Måler koblingen til unike klasser, gjennom å måle antall argumenter, returverdier, metodekall osv. som benyttes fra klassene. Dersom det er mange koblinger kan det bli vanskelig å gjenbruke deler av koden.

#### **Lines of code (LC)**

Antall kodelinjer, beregnet i *Intermediate Language (IL)* (tidligere kjent som *MSIL*). Antallet svarer ikke nødvendigvis til antallet linjer i *C#*-koden, men enkelt sagt kan man si at det er antall kodelinjer i programmet. Blanke linjer og kommentarlinjer er ikke inkludert.



## G.4.2 Nøkkeltall for versjon 3.0 og 3.1

Tabell G.2: *Code metrics* - versjon 3.0

Prosjekt	MI	CC	DI	CCo	LC
fap2D	61	2	1	8	12
fap2D.Computations	76	2539	5	114	6958
fap2D.Design	82	34	1	1	61
fap2D.Gui	57	5631	10	599	38992
fap2D.ModelData	83	252	9	94	779
fap2D.Storing	97	477	4	249	2210
fap2D.StructuralData	85	4117	4	128	5926
fap2D.Utilities	83	681	7	98	1757

Tabell G.3: *Code metrics* - versjon 3.1

Prosjekt	MI	CC	DI	CCo	LC
fap2D	61	2	1	8	12
fap2D.Computations	76	2581	5	119	7134
fap2D.Design	82	34	1	1	61
fap2D.Gui	57	6281	10	623	42702
fap2D.ModelData	83	252	9	94	779
fap2D.Storing	97	575	4	250	2357
fap2D.StructuralData	85	4203	4	130	6059
fap2D.Utilities	82	690	7	101	1822

## G.4.3 Oppsummering

### Maintainability index

Indeksen er omtrent uforandret og det er bra, siden verdiene er relativt gode.

### Cyclomatic complexity

Kompleksiteten har økt en del for fap2D.Gui, fap2D.Storing og fap2D.StructuralData. Det er naturlig for det er disse som har de største inngrepene. Nye *settings* andre funksjoner bidrar til enda flere stier i programmet.

### Class coupling

Koblingene har økt, men ikke vesentlig mye.

### Depth of inheritance

Hierarkiet er som forventet, akkurat som før. Det har ikke blitt lagt spesielt mange nye klasser.

### Lines of code

Antall kodelinjer har økt i de fleste prosjektene. Det er naturlig siden man ikke har fokusert på å trimme bort kode. fap2D.Gui har vokst desidert mest og er også mye større en de andre fra før av.

# Tillegg H

## Innhold på vedlagt minnepenn

Forklaring til innholdet på vedlagt minnepenn.

### H.1 Rapport

I denne mappen ligger rapporten i masteroppgaven, i PDF-format.

### H.2 Installasjonsfiler

I denne mappen ligger installasjonsfilene til den nye versjonen, **fap2D 3.1**, og forrige versjon, **fap2D 3.0.0 Beta**.

I tillegg ligger andre installasjonsfiler som trengs for å få installert og kjørt **fap2D**.

### H.3 Kildekode

I denne mappen ligger en zippet mappe med kildekoden til **fap2D 3.1**.

### H.4 Referanser

I denne mappen ligger noen dokumenter som det refereres til i rapporten.

- *User's manual*, brukermanualen til **fap2D**, i PDF-format. [7]
- Rapportene fra Espen Skogsruds prosjektoppgaver høsten 2013, i PDF-format. [11]
- Rapportene fra Torjus Sandvikens prosjektoppgaver høsten 2013, i PDF-format. [10]