

Prosjektoppgave i numerikk - TDAT2002

Tobias Andersen, Kjerand Evje, Dilawar Mahmood, Håvard Markhus

Våren 2020

Innholdsfortegnelse

1	Innledning	3
2	Konstanter og funksjoner	4
3	Teori	5
3.1	Euler-Bernoullibjelken	5
3.2	Taylors teorem med restledd	6
3.3	Skjæringssetningen	6
3.4	Definisjon av den deriverte	7
3.5	Vertikal forskyvning	7
3.6	Strukturmatrisen	8
3.7	Kondisjonstall	9
3.8	Maskinepsilon	10
3.9	Randbetingelser	10
4	Resultater	11
4.1	Oppgave 1	11
4.1.1	Deloppgave 5.1.21	11
4.1.2	Deloppgave 5.1.22	12
4.2	Oppgave 2	14
4.3	Oppgave 3	15
4.4	Oppgave 4	16
4.5	Oppgave 5	20
4.6	Oppgave 6	21
4.7	Oppgave 7	27
5	Konklusjon	28
5.1	Videre arbeid	29
6	Referanseliste	30
7	Vedlegg	30
7.1	Oppgave 2	30
7.2	Oppgave 3	30
7.3	Oppgave 4	31
7.4	Oppgave 5	33
7.5	Oppgave 6	34
7.6	Oppgave 7	35

1 Innledning

Denne rapporten er en del av vurderingen i numerikkdelen av faget TDAT2002. Rapporten tar for seg Euler-Bernoulli bjelken, en matematisk modell som forteller hvordan et materiale bøyes avhengig av lasten som benyttes. Vi har gjort beregninger og bevist ulike likninger rundt denne modellen, samt sett på hvordan kondisjonstall og maskinepsilon kan gi unøyaktigheter i svaret.

Vi har valgt å dele rapporten opp i følgende delkapitler:

- **Konstanter og funksjoner:** En oversikt over konstanter og funksjoner som er knyttet til Euler-Bernoulli bjelken.
- **Teori:** Dette delkapittelet ser på ulike teoremer, definisjoner og likninger som legger grunnlaget for delkapittel 3, Resultat.
- **Resultater:** Her vises det utregninger tilknyttet Euler-Bernoulli bjelken, og hvordan disse utregningene er basert på teoridelen av rapporten. Disse beregningene har blitt gjort i MATLAB og Python.
- **Konklusjon:** I dette delkapittelet oppsummerer vi funnene vi har gjort i rapporten og hvordan disse resultatene henger sammen for å gi et helhetlig bilde av Euler-Bernoulli bjelken. Her har vi også nevnt videre arbeid som kan gjøres for å gi en mer presis modell.
- **Referanser:** En liste over referanser som har blitt brukt i rapporten.
- **Vedlegg:** Kildekode som har blitt brukt til å produsere resultatene i rapporten.

2 Konstanter og funksjoner

D	Tettheten til stupebrettet	480 kg/m^3
d	Tykkelsen til stupebrettet	0.03 m
E	Elastisitetsmodul	$1.3 \cdot 10^{10} \text{ N/m}^2$
f	Massen til stupebrettet fordelt over lengden	$\approx -42.3792 \text{ N/m}$
$f(x)$	Masse fordelt over stupebrettet inkluder $s(x)$	$f + \sum_{i=1}^n s_i(x)$
g	Tyngdeakselerasjon	$\approx 9.81 \text{ m/s}^2$
I	Arealtreghetsmoment	$\approx 6.75 \cdot 10^{-7} \text{ m}^4$
L	Lengden til stupebrettet	2 m
p	Masse per meter	100 kg/m
$s(x)$	Sinusformet haug	$-pg \sin \frac{\pi}{L}x$
w	Bredden til stupebrettet	0.3 m
$y(x)$	Den vertikale forskyvningen til stupebrettet	$-pg \sin \frac{\pi}{L}x$

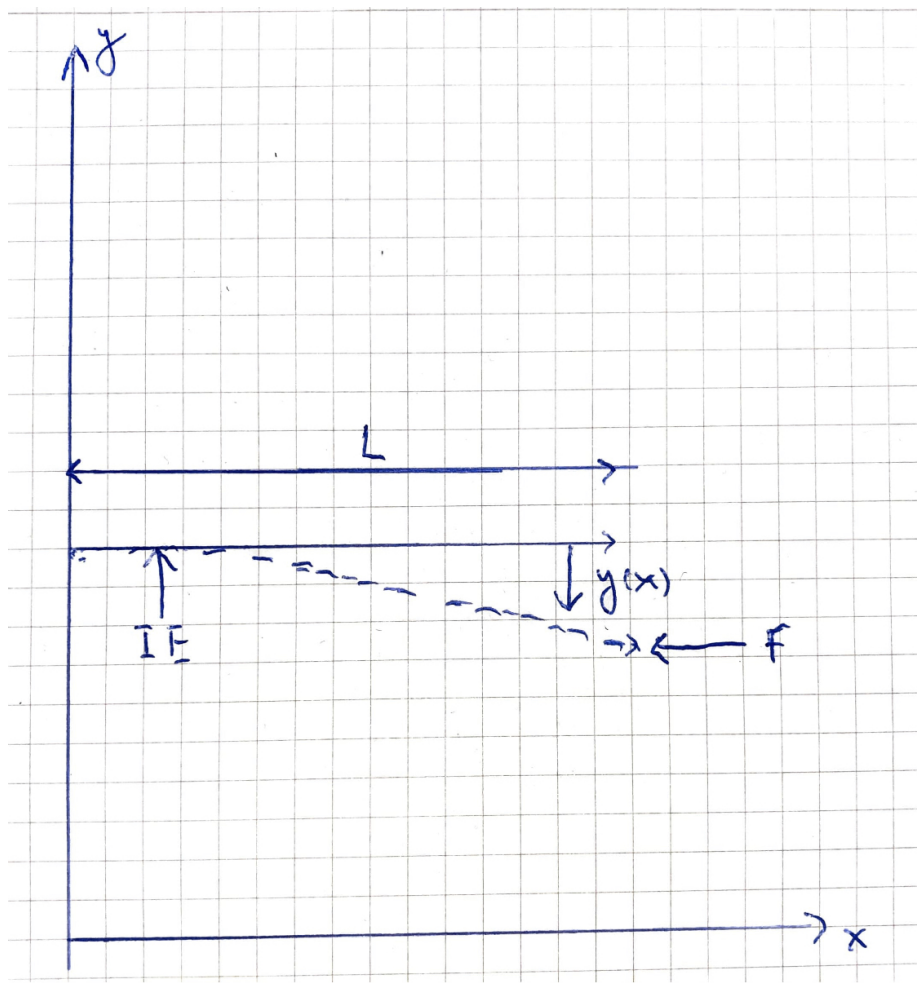
3 Teori

3.1 Euler-Bernoullibjelken

Euler-Bernoullibjelken er en modell som beskriver forskyvningen for materialer under belastning, inkludert egenmassen til bjelken. Euler-Bernoulli-likningen er gitt ved:

$$EIy'''' = f(x)$$

Se kapittel 2 for beskrivelsen av de ulike konstantene og funksjonene som er brukt i denne likningen. Nedenfor følger det en skisse som viser hvordan bjelken ser ut før og etter forskyvningen, beskrevet vha. Euler-Bernoulli-likningen:



3.2 Taylors teorem med restledd

La x og x_0 være reelle tall, og la f være $k + 1$ ganger kontinuerlig deriverbar i intervallet mellom x og x_0 . Da eksisterer det et tall c mellom x og x_0 slik at

$$f(x) = \left(\sum_{i=0}^k \frac{f^{(i)}(x_i)}{i!} (x - x_i)^i \right) + \frac{f^{(k+1)}(c)}{(k+1)!} (x - x_0)^{k+1}.$$

Summen opp til k kalles for Taylorpolynomet av grad k for f sentrert om punktet x_0 . Det siste leddet er restleddet. Hvis restleddet er liten, så vil Taylors Teorem gjøre det enkelt å approksimere en generell og glatt funksjon med et polynom.

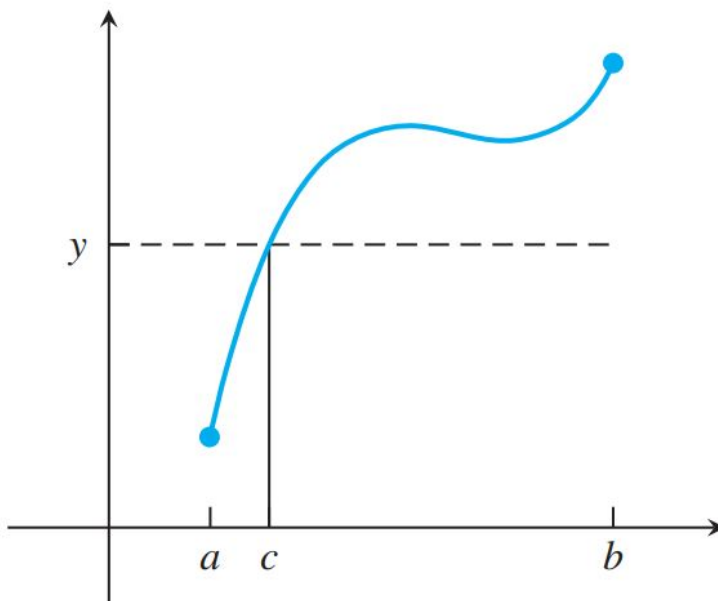
Restleddet er gitt ved funksjonen

$$R(x)_k = \frac{f^{(k+1)}(c)}{(k+1)!} (x - x_0)^{k+1}$$

Restleddet angir feilen fra den faktiske funksjonen og Taylorpolynomet. Når $k \rightarrow \infty$, så kommer $R(x)_k \rightarrow 0$. Det betyr at jo flere ledd vi legger til i Taylorsummen, jo bedre blir estimatet for den faktiske funksjonen.

3.3 Skjæringssetningen

La f være en kontinuerlig funksjon i intervallet $[a, b]$. Da vil den treffe alle verdier mellom $f(a)$ og $f(b)$. En mer presis definisjon er at hvis y er et tall slik at $f(a) \leq y \leq f(b)$, så eksisterer det et tall c hvor $a \leq c \leq b$ slik at $f(c) = y$.



3.4 Definisjon av den deriverte

Uttrykket som definerer den deriverte er *Newton's difference quotient* (brøken under) og sier at grenseverdien for differenskvotienten, med hensyn på h representerer stigningstallet for tangenten i punktet $(x, f(x))$. Dette gjelder hvis grenseverdien eksisterer.

$f'(x)$ er altså en funksjon for stigningstallet for $f(x)$

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

3.5 Vertikal forskyvning

Fra teori del 3.1 om Euler-Bernoulli har man følgende ligning.

$$EI y'''' = f(x)$$

Fra seksjon 4.1 har man også tilnærmingen for $f''''(x)$ av forskyvnings funksjonen til bjelken.

$$y'''' \approx \frac{y(x-2h) - 4y(x-h) + 6y(x) - 4y(x+h) + y(x+2h)}{h^4}$$

Setter sammen disse ligningene og får følgende uttrykk.

$$\frac{f(x)h^4}{EI} = y(x-2h) - 4y(x-h) + 6y(x) - 4y(x+h) + y(x+2h)$$

Videre ønsker vi å gjøre om ligningen til å uttrykke for $f(x_i)$. Her vil ligningen uttrykke kraften på forskyvningen for en gitt x_i . Bruker følgende uttrykk til å skrive om ligningen.

$$h = x_i - x_{i-1}$$

$$h = x_{i+1} - x_i$$

Bruker uttrykkene overfor til å skrive om uttrykket for forskyvningen ved å skrive om $(x-2h)$ til x_{i-2} , $(x-h)$ til x_{i-1} , $(x+h)$ til x_{i+1} og $(x+2h)$ til x_{i+2} .

$$\frac{f(x_i)h^4}{EI} = y(x_{i-2}) - 4y(x_{i-1}) + 6y(x_i) - 4y(x_{i+1}) + y(x_{i+2})$$

Til slutt brukes at $y(x_i) = y_i$ og får da den endelige formelen som angir den vertikale forskyvningen for Euler-Bernoulli bjelken for en gitt x -verdi.

$$\frac{f(x_i)h^4}{EI} = y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2}$$

3.6 Strukturmatrisen

$$\begin{bmatrix} 16 & -9 & \frac{8}{3} & -\frac{1}{4} & & & & & & \\ -4 & 6 & -4 & 1 & & & & & & \\ 1 & -4 & 6 & -4 & 1 & & & & & \\ & 1 & -4 & 6 & -4 & 1 & & & & \\ & & \dots & \dots & \dots & \dots & \dots & & & \\ & & & \dots & \dots & \dots & \dots & \dots & & \\ & & & & 1 & -4 & 6 & -4 & 1 & \\ & & & & & 1 & -4 & 6 & -4 & 1 \\ & & & & & & \frac{16}{17} & -\frac{60}{17} & \frac{72}{17} & -\frac{28}{17} \\ & & & & & & -\frac{12}{17} & \frac{96}{17} & -\frac{156}{17} & \frac{72}{17} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ y_{n-1} \\ y_n \end{bmatrix} = \frac{h^4}{EI} \begin{bmatrix} f(x_1) \\ f(x_2) \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ f(n-1) \\ f(n) \end{bmatrix}$$

Her ser man strukturmatrisen som er en tilnærming av $y''''(x)$ for Euler-Bernoulli bjelke formelen.

$$\frac{f(x_i)h^4}{EI} = y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2}$$

Fra seksjon 3.6 i teorien har vi utledet formelen overfor. Koeffisientene er $[1, -4, 6, -4, 1]$. Disse koeffisientene blir repetert gjennom strukturmatrisen, bortsett fra de to første og de to siste radene. Må regne ut en tilnærming for disse radene ettersom de ikke er definert for uttrykket.

$$\frac{f(x_i)h^4}{EI} = y_{-1} - 4y_0 + 6y_1 - 4y_2 + y_3$$

Regner ut koeffisientene for første rad ved å sette inn $i = 1$. Her vil ikke y_{-1} være definert, så vi vil være nødt til å bruke en alternativ tilnærming for x_1 . Bruker da beviset fra seksjon 5.1.22 med følgende uttrykk.

$$f^{(iv)}(x+h) = \frac{16f(x+h) - 9f(x+2h) + \frac{8}{3}f(x+3h) - \frac{1}{4}f(x+4h)}{h^4} + O(h^4)$$

Kan erstatte f med y og $x+h$ med x_1 . Dessuten når h blir liten så vil $O(h^4)$ bli veldig liten. Man kan da bruke tilnærmingen nedenfor.

$$y''''(x_1) \approx \frac{16f(x_1) - 9f(x_1+h) + \frac{8}{3}f(x_1+2h) - \frac{1}{4}f(x_1+3h)}{h^4}$$

Koeffisientene til dette uttrykket gir oss verdiene for rad 1, $[16, -9, \frac{8}{3}, -\frac{1}{4}]$.

For rad 2 vil y_{i-2} være den eneste som ikke er definert, ettersom når $i = 2$ så blir den y_0 . Resten av koeffisientene vil være de samme som for store deler av matrisen. Verdiene for rad 2 blir da $[-4, 6, -4, 1]$.

For de 2 siste radene har vi også verdier i uttrykket som ikke er definert. Her

regner man også disse ut med alternative tilnærminger. Bruker ligningene (2.32) og (2.33) på side 103 i Numerical Analysis boka.

$$y''''(x_{n-1}) \approx \frac{-28y_n + 72y_{n-1} - 60y_{n-2} + 16y_{n-3}}{17h^4}$$

$$y''''(x_n) \approx \frac{72y_n - 156y_{n-1} + 96y_{n-2} - 12y_{n-3}}{17h^4}$$

Hvis man da setter inn Euler-Bernoulli ligningen i disse formlene får man følgende uttrykk.

$$\frac{h^4}{EI} f(x_{n-1}) = \frac{16}{17}y_{n-3} - \frac{60}{17}y_{n-2} + \frac{72}{17}y_{n-1} - \frac{28}{17}y_n$$

$$\frac{h^4}{EI} f(x_n) = -\frac{12}{17}y_{n-3} + \frac{96}{17}y_{n-2} - \frac{156}{17}y_{n-1} + \frac{72}{17}y_n$$

Koeffisientene til disse uttrykkene angir verdiene for de 2 siste radene i matrisen. Altså $[\frac{16}{17}, -\frac{60}{17}, \frac{72}{17}, -\frac{28}{17}]$ og $[-\frac{12}{17}, \frac{96}{17}, -\frac{156}{17}, \frac{72}{17}]$.

Konklusjonen er at strukturmatrisen er en utledning av Euler-Bernoulli ligningen

$$\frac{1}{h^4}Ay = \frac{f(x)}{EI} = y''''$$

hvor A er strukturmatrisen. Strukturmatrisen viser til Euler-Bernoulli ligningen for hvert punkt x_i på bjelken med $x_n = L$ og avstand h .

3.7 Kondisjonstall

Kondisjonstallet sier noe om hvor mye utgangsverdiene for en funksjon kan forandre seg i forhold til forandringer i inngangsverdiene. om verdien $f(x)$ forandrer seg mye med lite forandring av verdien til x , vil kondisjonstallet være høyt.

Når det gjelder matriser regnes kondisjonstallet ut slik for en matrise A :

$$cond(A) = \|A\| \cdot \|A^{-1}\|$$

Hos matriser med tanke på en approksimasjon av en ligning $Ax = b$ vil kondisjonstallet si noe om hvor unøyaktig løsningen for x vil være. Kondisjonstallet er uavhengig av avrundingen til flyttalls-beregning. Dette betyr at kondisjonstallet ikke kan ta til rette for feil som kan oppstå ved avrundning.

Det samme som gjelder funksjoner gjelder også her. Kondisjonstallet sier noe om forandringen til b iforhold til x . Om forandringene er store for b ved små endringer av x , vil kondisjonstallet til A være høyt. Ved approksimasjon av x vil små forandringer av b gi mindre feil ved et lite kondisjonstall og motsatt for et stort kondisjonstall.

Problemer med lave kondisjonstall kalles bra kondisjonert, mens problemer med høye kondisjonstall kalles dårlig kondisjonert. En matrise med kondisjonstall 1 er det beste som er mulig og vil være bra kondisjonert. Det motsatte gjelder for en matrise med høyt kondisjonstall, f.eks når kondisjonstallet går mot uendelig, og vil da være dårlig kondisjonert.

3.8 Maskinepsilon

Maskinepsilon (ϵ_{mach}) er definert som distansen mellom 1 og det minste tallet større enn 1 som en maskin kan forholde seg til. Dette bør en være klar over når en gjør beregninger med flyttall på en datamaskin. Datatypen Double (standard beskrevet av IEEE) er veldig vanlig å bruke for slike beregninger, der $\epsilon_{\text{mach}} = 2^{-52}$.

Grunnen til at vi ser toerpotenser her følger klart av at datamaskinen alltid egentlig forholder seg til binære tall. Helt konkret ser vi 52 fordi Double setter av 52 bits til å beskrive mantissen. Matlab bruker i utgangspunktet Double for sine beregninger, våre beregninger kan altså aldri være mer nøyaktige enn ϵ_{mach} .

3.9 Randbetingelser

En randbetingelse er et krav som må være oppfylt. Randbetingelser med tanke på Euler-Bernoulli-modellen er:

$$y(0) = y'(0) = y''(L) = y'''(L) = 0$$

hvor $y(x)$ beskriver formen på en bjelke og L lengden på bjelken.

Vi har at $y(0) = 0$ siden det ikke forekommer vertikal forskyvning i den enden stupebrettet er festet.

$y'(0) = 0$ siden stupebrettet er festet ved punktet $x = 0$, og stupebrettet er helt rett der.

$y''(L) = 0$ siden enden av stupebrettet ikke skal være bøyd.

$y'''(L) = 0$ siden det ikke skal skje noen rykning i enden av stupebrettet.

4 Resultater

4.1 Oppgave 1

4.1.1 Deloppgave 5.1.21

Må bevis at 2. ordens formelen for den fjerdederiverte er gitt ved:

$$f^{(iv)}(x) = \frac{f(x-2h) - 4f(x-h) + 6f(x) - 4f(x+h) + f(x+2h)}{h^4} + O(h^2)$$

Bevis

Taylor's teorem sier oss at dersom f er 6 ganger derivervar og kontinuerlig, så blir Taylor ekspansjonen til $f(x-2h)$ og $f(x+2h)$ følgende:

$$\begin{aligned} f(x-2h) &= \left(\sum_{i=0}^5 \frac{(-2h)^i}{i!} f^{(i)}(x) \right) + \frac{4h^6}{45} f^{(vi)}(c_1) \\ &= f(x) - 2hf'(x) + 2h^2 f''(x) - \frac{4h^3}{3} f'''(x) + \frac{2h^4}{3} f^{(iv)}(x) - \frac{4h^5}{15} f^{(v)}(x) + \frac{4h^6}{45} f^{(vi)}(c_1) \end{aligned}$$

og

$$\begin{aligned} f(x+2h) &= \left(\sum_{i=0}^5 \frac{(2h)^i}{i!} f^{(i)}(x) \right) + \frac{4h^6}{45} f^{(vi)}(c_2) \\ &= f(x) + 2hf'(x) + 2h^2 f''(x) + \frac{4h^3}{3} f'''(x) + \frac{2h^4}{3} f^{(iv)}(x) + \frac{4h^5}{15} f^{(v)}(x) + \frac{4h^6}{45} f^{(vi)}(c_2) \end{aligned}$$

Vi får at:

$$f(x-2h) + f(x+2h) = 2f(x) + 4h^2 f''(x) + \frac{4h^4}{3} f^{(iv)}(x) + \frac{8h^6}{45} f^{(vi)}(c_3) \quad (1)$$

Taylor ekspansjonen til $f(x-h)$ og $f(x+h)$ blir følgende:

$$\begin{aligned} f(x-h) &= \left(\sum_{i=0}^5 \frac{(-h)^i}{i!} f^{(i)}(x) \right) + \frac{h^6}{720} f^{(vi)}(c_4) \\ &= f(x) - hf'(x) + \frac{h^2}{2} f''(x) - \frac{h^3}{6} f'''(x) + \frac{h^4}{24} f^{(iv)}(x) - \frac{h^5}{120} f^{(v)}(x) + \frac{h^6}{720} f^{(vi)}(c_4) \end{aligned}$$

og

$$\begin{aligned} f(x+h) &= \left(\sum_{i=0}^5 \frac{h^i}{i!} f^{(i)}(x) \right) + \frac{h^6}{720} f^{(vi)}(c_5) \\ &= f(x) + hf'(x) + \frac{h^2}{2} f''(x) + \frac{h^3}{6} f'''(x) + \frac{h^4}{24} f^{(iv)}(x) + \frac{h^5}{120} f^{(v)}(x) + \frac{h^6}{720} f^{(vi)}(c_5) \end{aligned}$$

Vi får at:

$$\begin{aligned} f(x-h) + f(x+h) &= 2f(x) + h^2 f''(x) + \frac{h^4}{12} f^{(iv)}(x) + \frac{h^6}{360} f^{(vi)}(c_6) \implies \\ 4f(x-h) + 4f(x+h) &= 8f(x) + 4h^2 f''(x) + \frac{h^4}{3} f^{(iv)}(x) + \frac{h^6}{90} f^{(vi)}(c_6) \quad (2) \end{aligned}$$

Skriver om likningene (1) og (2):

$$(1) \quad 4h^2 f''(x) = f(x-2h) + f(x+2h) - 2f(x) - \frac{4h^4}{3} f^{(iv)}(x) - \frac{8h^6}{45} f^{(vi)}(c_3)$$

$$(2) \quad 4h^2 f''(x) = 4f(x-h) + 4f(x+h) - 8f(x) - \frac{h^4}{3} f^{(iv)}(x) - \frac{h^6}{90} f^{(vi)}(c_6)$$

Dette gir oss følgende likhet:

$$f(x-2h) + f(x+2h) - 2f(x) - \frac{4h^4}{3} f^{(iv)}(x) - \frac{8h^6}{45} f^{(vi)}(c_3) = 4f(x-h) + 4f(x+h) - 8f(x) - \frac{h^4}{3} f^{(iv)}(x) - \frac{h^6}{90} f^{(vi)}(c_6)$$

Skriver om:

$$h^4 f^{(iv)}(x) = f(x-2h) - 4f(x-h) - 2f(x) + 8f(x) - 4f(x+h) + f(x+2h) + O(h^6) \implies$$

$$f^{(iv)}(x) = \frac{f(x-2h) - 4f(x-h) + 6f(x) - 4f(x+h) + f(x+2h)}{h^4} + O(h^2) \quad \blacksquare$$

4.1.2 Deloppgave 5.1.22

La $f(x)$ være en seks ganger kontinuerlig deriverbar funksjon. Må bevise at hvis $f(x) = f'(x) = 0$, så vil

$$f^{(iv)}(x+h) - \frac{16f(x+h) - 9f(x+2h) + \frac{8}{3}f(x+3h) - \frac{1}{4}f(x+4h)}{h^4} = O(h^2) \quad (1)$$

Bevis

Starter med å bevise følgende likhet:

$$f(x-h) - 10f(x+h) + 5f(x+2h) - \frac{5}{3}f(x+3h) + \frac{1}{4}f(x+4h) = O(h^6) \quad (2)$$

Bruker Taylors teorem til å utlede uttrykk for $f(x+3h)$ og $f(x+4h)$:

$$f(x+3h) = \left(\sum_{i=0}^5 \frac{(3h)^i}{i!} f^{(i)}(x) \right) + \frac{81h^6}{80} f^{(vi)}(c_3)$$

$$= f(x) + 3hf'(x) + \frac{9}{2}h^2 f''(x) + \frac{9h^3}{28} f'''(x) + \frac{27h^4}{8} f^{(iv)}(x) + \frac{81h^5}{40} f^{(v)}(x) + \frac{81h^6}{80} f^{(vi)}(c_3)$$

og

$$f(x+4h) = \left(\sum_{i=0}^5 \frac{(4h)^i}{i!} f^{(i)}(x) \right) + \frac{256h^6}{45} f^{(vi)}(c_4)$$

$$= f(x) + 4hf'(x) + 8h^2 f''(x) + \frac{32h^3}{3} f'''(x) + \frac{32h^4}{3} f^{(iv)}(x) + \frac{128h^5}{15} f^{(v)}(x) + \frac{256h^6}{45} f^{(vi)}(c_4)$$

Ved å multiplisere uttrykkene for $f(x-h)$, $f(x+h)$, $f(x+2h)$, $f(x+3h)$ og $f(x+4h)$ med henholdsvis 1, -10, 5, $-\frac{5}{3}$ og $\frac{1}{4}$, og å summere dem sammen samtidig som man bruker at $f(x) = f'(x) = 0$, så får vi at:

$$f(x-h) - 10f(x+h) + 5f(x+2h) - \frac{5}{3}f(x+3h) + \frac{1}{4}f(x+4h) = O(h^6)$$

Fra 5.1.21 har vi at:

$$f^{(iv)}(x) = \frac{f(x-2h) - 4f(x-h) + 6f(x) - 4f(x+h) + f(x+2h)}{h^4} + O(h^2)$$

Da blir uttrykket for $f(x+h)$ følgende:

$$f^{(iv)}(x+h) = \frac{f(x-h) - 4f(x) + 6f(x+h) - 4f(x+2h) + f(x+3h)}{h^4} + O(h^2)$$

Setter inn uttrykket i (1):

$$f^{(iv)}(x+h) - \frac{16f(x+h) - 9f(x+2h) + \frac{8}{3}f(x+3h) - \frac{1}{4}f(x+h)}{h^4} = O(h^2)$$

$$\frac{f(x-h) - 4f(x) + 6f(x+h) - 4f(x+2h) + f(x+3h) - 16f(x+h) + 9f(x+2h) + \frac{8}{3}f(x+3h) - \frac{1}{4}f(x+h)}{h^4} = O(h^2)$$

Setter $f(x) = 0$, og får til slutt at:

$$\frac{f(x-h) - 10f(x+h) + 5f(x+2h) - \frac{5}{3}f(x+3h) + \frac{1}{4}f(x+4h)}{h^4} = O(h^2) \implies$$

$$f(x-h) - 10f(x+h) + 5f(x+2h) - \frac{5}{3}f(x+3h) + \frac{1}{4}f(x+4h) = O(h^6)$$

Vi ender opp med likning (2), noe som betyr at (1) stemmer hvis

$$f(x) = f'(x) = 0 \quad \blacksquare$$

4.2 Oppgave 2

Skal lage et program i MatLab for å definere strukturmatrisen gitt i (2.34).

```
1 % Oppgave 2
2 function A = lagA(n)
3     e = ones(n,1);
4     A = spdiags([e, -4*e, 6*e, -4*e, e], [-2, -1, 0, 1, 2], n, n);
5
6     A(1, 1:4) = [16, -9, 8/3, -1/4];
7     A(n-1, n-3:n) = [16/17, -60/17, 72/17, -28/17];
8     A(n, n-3:n) = [-12/17, 96/17, -156/17, 72/17];
9 end
```

Funksjonen skal for et gitt antall n produsere en $n \times n$ strukturmatrise. Tar utgangspunkt i kodesnutten gitt i oppgaveteksten og lager den følgende funksjonen. Setter inn verdiene for rad 1, $n-1$ og n basert på strukturmatrisen fra (2.34).

```
1  □ function A = lagA(n)
2  -     e = ones(n);
3  -     A = spdiags([e, -4*e, 6*e, -4*e, e], [-2, -1, 0, 1, 2], n, n);
4  -
5  -     A(1, 1:4) = [16, -9, 8/3, -1/4];
6  -     A(n-1, n-3:n) = [16/17, -60/17, 72/17, -28/17];
7  -     A(n, n-3:n) = [-12/17, 96/17, -156/17, 72/17];
8  -     end
```

4.3 Oppgave 3

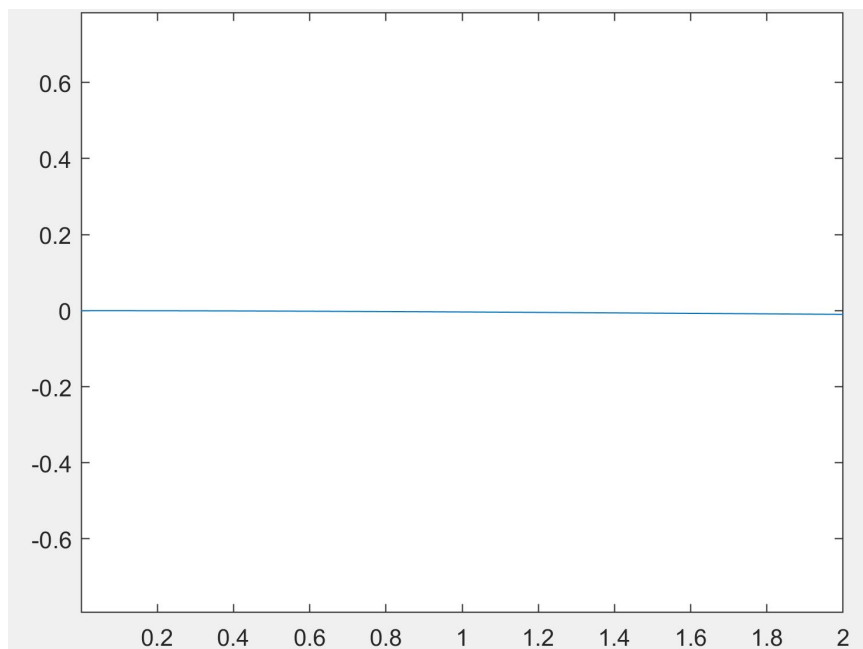
Skal finne løsningen y_c på problemet med en bjelke uten ekstra masse enn egenmassen ved å bruke $n = 10$ intervaller. La A være matrisen fra oppgave 2. Siden $n = 10$:

$$h = \frac{L}{n} = \frac{2}{10} = 0.2$$
$$b = \left[\frac{h^4}{E \cdot I} \cdot f \right]_{n \times 1}$$

Skal løse likningen $Ay = b$ for å finne løsningen på problemet. Se matlabkoden for oppgave 3. Vi får at

$$y = \begin{bmatrix} -0.000180624738461544 \\ -0.000674847507692328 \\ -0.00141698658461543 \\ -0.00234908750769237 \\ -0.00342092307692317 \\ -0.00458999335384627 \\ -0.0058215256615386 \\ -0.00708847458461555 \\ -0.00837152196923095 \\ -0.00965907692307711 \end{bmatrix}$$

Dette er forskyvningen stupebrettet har pga. sin egenmasse. Nedenfor er et bilde av hvordan denne bøyningen ser ut, hvor x-aksen er lengden i meter, og y-aksen er forskyvningen i meter.



4.4 Oppgave 4

a)

$$\begin{aligned}y(x) &= \left(\frac{f}{24EI}\right)x(x - 4Lx + 6L) \\y'(x) &= \frac{fx(x - 3Lx + 3L)}{6EI} \\y''(x) &= \frac{f(x - L)^2}{2EI} \\y'''(x) &= \frac{f(x - L)}{EI} \\y''''(x) &= \frac{f}{EI}\end{aligned}$$

Setter inn y'''' i euler ligningen:

$$\begin{aligned}EIy'''' &= f(x) \\ \frac{fEI}{IE} &= f(x) \\ \Rightarrow \underline{\underline{f = f(x)}}$$

b) Skal vise at $y^{(6)}(c) = 0$:

Vet at $y''''(x) = \frac{f}{EI}$ fra forrige deloppgave, regner derfor ut $y^{(5)}$ og $y^{(6)}$

$$\begin{aligned}y^{(5)}(x) &= 0 \\ \Rightarrow y^{(6)}(x) &= 0\end{aligned}$$

Dette gjør at $\frac{y^{(6)}(c)}{6!}h^6 = 0$ som gjør at den 4-deriverte $\frac{1}{h^4}Ay$ er eksakt

c) Skal regne ut vektoren $y_e = [y(0.2), y(0.4), y(0.6) \dots y(2.0)]^T$ for den eksakte løsningen

$$y(x) = \left(\frac{f}{24EI}\right)x(x - 4Lx + 6L)$$

Trenger da konstantene f, E, I og L, hentet fra boka Numerical Analysis av Timothy Sauer¹.

Regner ut f ved hjelp av tykkelsen w og bredden d som er oppgitt i boka:

$$\begin{aligned}w &= 0.3m \\ d &= 0.03m \\ f &= -480wdg = -480 \cdot 0.3 \cdot 0.03 \cdot 9.81 = -42.3792\end{aligned}$$

Regner ut I ved hjelp av w og d som er oppgitt i boka:

$$I = \frac{wd^3}{12} = \frac{0.3 \cdot (0.03)^3}{12} = 6.75 \cdot 10^{-7}$$

$$E = 1.3 \cdot 10^{10} \frac{N}{m^2}$$

$$L = 2m$$

Regner ut y_e med python og får :

$$\begin{bmatrix} -0.0001806247384615385 \\ -0.0006748475076923078 \\ -0.0014169865846153848 \\ -0.002349087507692308 \\ -0.0034209230769230766 \\ -0.004589993353846155 \\ -0.005821525661538462 \\ -0.007088474584615385 \\ -0.00837152196923077 \\ -0.009659076923076923 \end{bmatrix}$$

Regner så ut den numeriske fjerdederiverte av $y(x)$ ved hjelp av den numeriske fjerdederiverte $\frac{1}{h^4}Ay$

Lager matrisen A ved hjelp av lagA() fra oppgave 2:

$$A = \begin{bmatrix} 16 & -9 & \frac{8}{3} & -\frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\ -4 & 6 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 6 & -4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -4 & 6 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{16}{17} & \frac{-60}{17} & \frac{72}{17} & \frac{-28}{17} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-12}{17} & \frac{96}{17} & \frac{-156}{17} & \frac{72}{17} \end{bmatrix}$$

Da blir $\frac{1}{h^4} Ay = \frac{1}{0.2^4} Ay :$

$$\begin{bmatrix} -0.004829538461538441 \\ -0.004829538461538441 \\ -0.004829538461537525 \\ -0.004829538461541549 \\ -0.004829538461531779 \\ -0.004829538461542882 \\ -0.004829538461534888 \\ -0.004829538461535776 \\ -0.004829538461541105 \\ -0.004829538461532234 \end{bmatrix}$$

Regnet ut i python, vedlegg 4.py

d) Skal sammenligne den numeriske fjerdederiverte vi fant i oppgave c) med vektoren $\frac{f}{EI} [1 \ 1 \ 1 \ \dots \ 1]^T$ og skal finne foroverfeil og relativ foroverfeil til den numeriske fjerdederiverte $\frac{1}{h^4} Ay_e$. Antar at relativ bakoverfeil er e . Regner først ut $\frac{f}{EI} [1 \ 1 \ 1 \ \dots \ 1]^T$ og bruker konstantene fra de tidligere oppgavene:

$$\frac{-42.3792}{1.3 \cdot 10^{10} \cdot 6.75 \cdot 10^{-7}} \cdot [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]^T = \begin{bmatrix} -0.0048295384615384615 \\ -0.0048295384615384615 \\ -0.0048295384615384615 \\ -0.0048295384615384615 \\ -0.0048295384615384615 \\ -0.0048295384615384615 \\ -0.0048295384615384615 \\ -0.0048295384615384615 \\ -0.0048295384615384615 \\ -0.0048295384615384615 \end{bmatrix}$$

Ser at vektoren er meget lik numerisk fjerdederivert $\frac{1}{h^4} Ay_e$. Finner nå foroverfeil med formel:

$$\|x - x_0\|_\infty$$

Trenger da kun å finne den største differansen for hvert ledd i de to vektorene. Regner det ut med python (vedlegg 4.py) og differansen på vektorene blir da slik:

$$\begin{bmatrix} 2.0816681711721685e-17 \\ 2.0816681711721685e-17 \\ 9.08995101411847e-16 \\ 3.0878077872387166e-15 \\ 6.682154829462661e-15 \\ 4.4200754167889045e-15 \\ 3.573530360512223e-15 \\ 2.6853519408120974e-15 \\ 2.643718577388654e-15 \\ 6.238065619612598e-15 \end{bmatrix}$$

Ser at største differanse er $6.682154829462661 \cdot 10^{-15}$

Relativ foroverfeil beregnes med formel:

$$\frac{\|x - x_0\|_{\infty}}{\|x\|_{\infty}}$$

Relativ foroverfeil blir da:

$$\frac{6.682154829462661 \cdot 10^{-15}}{0.0048295384615384615} = \underline{\underline{1.383601121034254 \cdot 10^{-12}}}$$

Feilforstørring: relativ bakoverfeil er gitt i oppgaven til 2^{-52}

$$Feilforstørring = \frac{relativforoverfeil}{relativbakoverfeil} = \frac{1.383601121034254 \cdot 10^{-12}}{2^{-52}} = \underline{\underline{6231.1854931192665}}$$

Tallet beskriver forholdet mellom relativ bakoverfeil og relativ foroverfeil

Feilforstørringen skal sammenlignes med kondisjonstallet til A. Må da først finne kondisjonstallet. Bruker lagA fra oppgave 2 og $\text{cond}(A)$ i matlab og får svaret:

$$\text{cond}(A) = 33254$$

Kondisjonstallet forteller hva som er maksimalt mulig feilforstørringsfaktor. Feilforstørringsfaktoren vår er 6231 og er betydelig mindre enn 33254. dette betyr at resultatet vi har fått er en god tilnærming.

4.5 Oppgave 5

Kjører utregningene i oppgave 3 om igjen for $n = 10, 40, \dots, 10 \cdot 2^{11}$. Nedenfor er det en tabell over feilen i punktet $x = L$ for hver n , samt kondisjonstallet til A for de forskjellige verdiene av n :

n	Feil	Kondisjonstall
20	1.3761561334924e-14	530302.499998998
40	1.47331799538186e-13	8449279.99985676
80	6.66359328826971e-13	134821259.980659
160	-8.60994955886873e-12	2153877317.45886
320	1.72876280998824e-10	34434645510.4922
640	-8.31952883320608e-10	550730030029.797
1280	-2.2284891572702e-08	8809861144987.75
2560	1.03881457609589e-07	140942568348747
5120	1.66558323965414e-06	2.25488634226904e+15
10240	-4.36949342139947e-05	3.60725725331037e+16
20480	-3.18512163212936e-05	5.77022291760934e+17

Vi ser fra tabellen at den absolutte feilen er størst for $n = 10240$. Feilen endrer seg med n som den gjør fordi kondisjonstallet til A blir større når n blir større. Dette medfører at sjansen for avrundingsfeil øker.

4.6 Oppgave 6

a)

$$f = -480wdg$$

Siden det legges en sinusformet haug på stupebrettet må vi legge på funksjonen

$$s(x) = -pg\sin\left(\frac{\pi}{L}x\right)$$

på f :

$$f(x) = -480wdg - pg\sin\left(\frac{\pi}{L}x\right)$$

Skal i oppgaven vise at

$$y(x) = \frac{f}{24EI}x^2(x^2 - 4Lx + 6L^2) - \frac{gpL}{EI\pi}\left(\frac{L^3}{\pi^3}\sin\left(\frac{\pi}{L}x\right) - \frac{x^3}{6} + \frac{Lx^2}{2} - \frac{L^2x}{\pi^2}\right)$$

tilfredsstiller Euler-Bernoulli-likningen og randbetingelsene for en bjelke som er festet i den ene enden og fri i den andre.

Starter da med å derivere $y(x)$ 4 ganger (bruker symbolab for utregning):

$$\begin{aligned} y'(x) &= \frac{fx(x^2 - 3Lx + 3L^2)}{6EI} - \frac{Lgp(-\pi^2x^2 + 2\pi^2Lx + 2L^2\cos(\frac{\pi x}{L}) - 2L^2)}{2\pi^3EI} \\ y''(x) &= \frac{2gL^2p\sin(\frac{\pi x}{L}) + \pi^2fx^2 + (2\pi gLp - 2\pi^2fL)x - 2\pi gL^2p + \pi^2fL^2}{2\pi^2EI} \\ y'''(x) &= \frac{gLp\cos(\frac{\pi x}{L}) + \pi fx + gLp - \pi fL}{\pi EI} \\ y''''(x) &= -\frac{gpsin(\frac{\pi x}{L}) - f}{EI} = \frac{f - gpsin(\frac{\pi x}{L})}{EI} \end{aligned}$$

Sjekker med Euler-Bernoulli-likningen:

$$EIy'''' = EI \cdot \frac{f - gpsin(\frac{\pi x}{L})}{EI} = \underline{\underline{f - gpsin(\frac{\pi x}{L}) = f(x)}}$$

Dette viser at $y(x)$ oppfyller Euler-bernoulli-likningen.

Neste steg er å vise at den oppfyller rand-betingelsene:

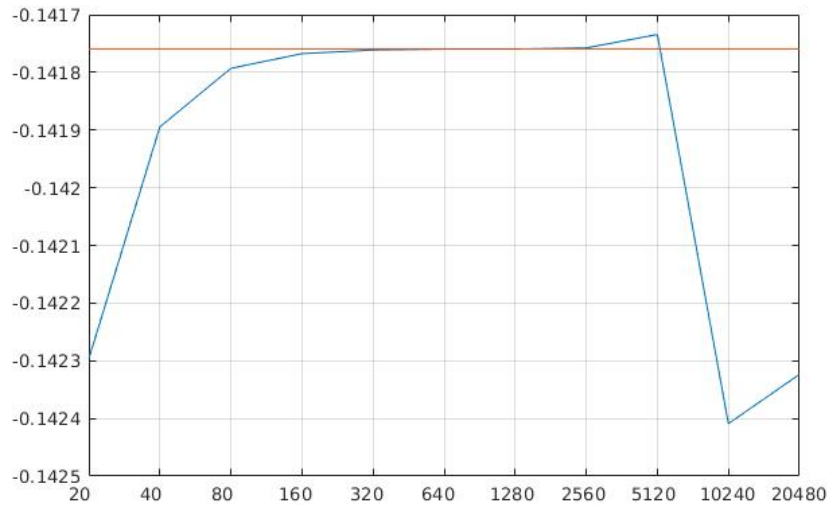
$$y(0) = y'(0) = y''(L) = y'''(L) = 0$$

Sjekker for $y(0), y'(0), y''(L), y'''(L)$:

$$\begin{aligned}
 y(0) &= \frac{f}{24EI} \cdot 0(0 - 4L \cdot 0 + 6L^2) - \frac{gpL}{EI\pi} \left(\frac{L^3}{\pi^3} \sin\left(\frac{\pi}{L}0\right) - \frac{0^3}{6} + \frac{L \cdot 0^2}{2} - \frac{L^2 \cdot 0}{\pi^2} \right) = 0 \\
 y'(0) &= \frac{f0(0^2 - 3L0 + 3l^2)}{6EI} - \frac{Lgp(-\pi^2 \cdot 0^2 + 2\pi^2 L0 + 2L^2 \cos(\frac{\pi \cdot 0}{L}) - 2L^2)}{2\pi^3 EI} = 0 \\
 y''(L) &= \frac{2gL^2 p \sin(\frac{\pi L}{L}) + \pi^2 f L^2 + (2\pi g L p - 2\pi^2 f L)L - 2\pi g L^2 p + \pi^2 f L^2}{2\pi^2 EI} \\
 &= \frac{2\pi^2 f L^2 - 2\pi^2 f L^2 + 2\pi g L^2 p - 2\pi g L^2 p}{2\pi^2 EI} = \frac{0 + 0}{2\pi^2 EI} = 0 \\
 y'''(L) &= \frac{gLp \cos(\frac{\pi L}{L}) + \pi f L + gLp - \pi f L}{\pi EI} = \frac{-gLp + gLp - \pi f L + \pi f L}{\pi EI} = 0
 \end{aligned}$$

Ser da at $y(x)$ oppfyller randbetingelsene.

b) Skal gjøre det samme som i oppgave 5, men med den sinusformede haugen i tillegg. Derfor blir $y(x)$ og $f(x)$ annerledes. Rød strek er den riktige verdien for $y(x)$, mens den blå linja viser den numeriske bergnede løsningen. Feilen er stor i starten når antallet n er liten (ved $A = 20 \times 20$ matrise) og blir mer lik riktig verdi når n går mot 320. Feilen blir større igjen når n blir veldig høy (etter 2560)



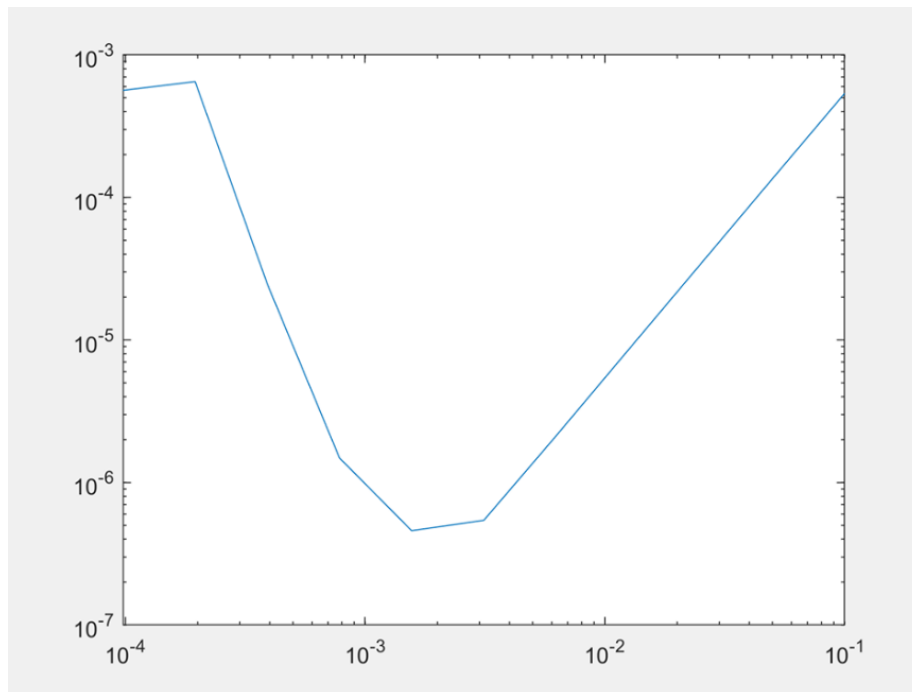
x-akse er n for $\text{lagA}(n)$. y-akse er siste verdi av $y(x)$ for gitt n . Graf laget i MATLAB, i vedlegg oppg6b.m

c)

Lager en graf for feilen for y_n i oppgave 6b). Bruker loglog kommandoen i MatLab til å lage en graf med logaritmisk skala.

```
loglog(arrayfun(@(n) L/n, n_s), arrayfun(@(x) abs(x), feil));  
hold on;
```

Vil da få følgende graf for feilen:



Denne grafen viser feilen for ulike verdier av n . Her viser x-aksen stegstørrelsen (lengden av bjelken delt på n) og y-aksen viser feilen. Her ser man at feilen blir mindre når n øker, helt frem til $n = 1280$. Etter dette punktet så stiger feilen igjen når n fortsetter å bli større.

n	Feil	Kondisjonstall
20	0.0005377	5.303e+05
40	0.00013546	8.4493e+06
80	3.3931e-05	1.3482e+08
160	8.4869e-06	2.1539e+09
320	2.1194e-06	3.4435e+10
640	5.4294e-07	5.5073e+11
1280	4.5913e-07	8.8099e+12
2560	1.4881e-06	1.4094e+14
5120	2.4717e-05	2.2549e+15
10240	0.00064995	3.6073e+16
20480	0.00056448	5.7702e+17

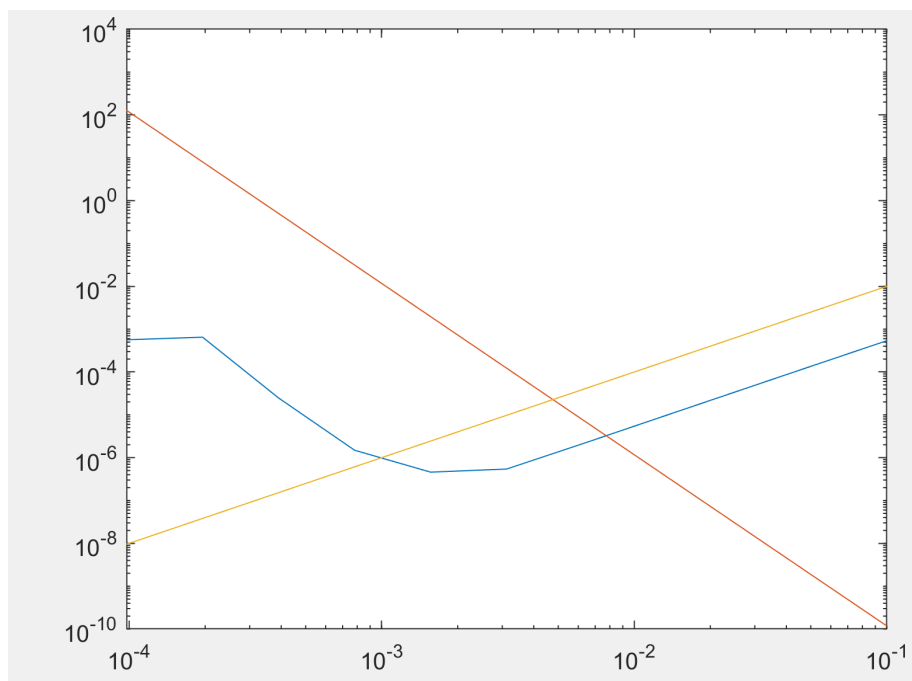
Denne tabellen viser feil og kondisjonstall for noen n-verdier. Her ser man at feilen blir mindre helt frem til $n = 1280$, når feilen begynner å øke igjen. Dette er fordi når n blir veldig stor, så vil også kondisjonstallet bli stort og man vil dermed få større usikkerhet i beregningen, og dermed større feil.

d)

Lager en graf som inneholder den teoretiske feilen, kondisjonstallet til A multiplisert med ϵ_{mach} , samt funksjonen fra 6c). Bruker følgende kode i MatLab:

```
teoretisk_feil = arrayfun(@(x) L^2 / x^2, n_s);
j = kond * eps(1);
loglog(arrayfun(@(n) L/n, n_s), theoretisk_feil);
hold on;
loglog(arrayfun(@(n) L/n, n_s), j);
```

Finner den teoretiske feilen og finner produktet av kondisjonstallet og ϵ_{mach} . Lager deretter en graf med logaritmisk skala. Får da følgende graf:



Denne grafen viser kondisjonstallet A multiplisert med ϵ_{mach} og den teoretiske feilen plottet sammen grafen fra forrige oppgave.

Her representerer den blå linjen den numeriske metoden fra oppgave c), den gule linjen representerer den teoretiske feilen og den røde linjen kondisjonstallet A multiplisert med ϵ_{mach} . Her ser man at i starten så vil den numeriske metoden likne mest på den røde linjen og være ca. parallell med den. Men etter man har passert punktet hvor $n = 1280$, så vil den numeriske metoden begynne å likne mer på og være parallell med den teoretiske feilen. Dette er forårsaket av at når n er liten så vil kondisjonstallet også være lite, som betyr at man har god sikkerhet i beregningene. Ettersom n blir større vil kondisjonstallet bli større, og man vil få større usikkerhet i beregningene. Den teoretiske feilen derimot vil bli mindre og mindre når n blir større. Dermed ser man at den numeriske metoden er mest lik produktet av kondisjonstallet A og ϵ_{mach} i starten, men likner mer på den teoretiske feilen når n blir større.

e)

Den teoretiske feilen er gitt som $h^2 = L^2/n^2$. L er en konstant, så den teoretiske feilen endres kun ved n . Siden man deler på n , så vil den teoretiske feilen bli mindre og mindre ettersom n blir større. Når n går mot uendelig så vil den teoretiske feilen gå mot null. Men selv om den teoretiske feilen fortsetter å minke ved stor n , så vil feilen fortsatt øke. Det er fordi kondisjonstallet A blir større ved store n , og dette medfører større usikkerhet i beregningene og større

feil. Den teoretiske feilen blir altså mindre og mindre, men den faktiske feilen øker.

f)

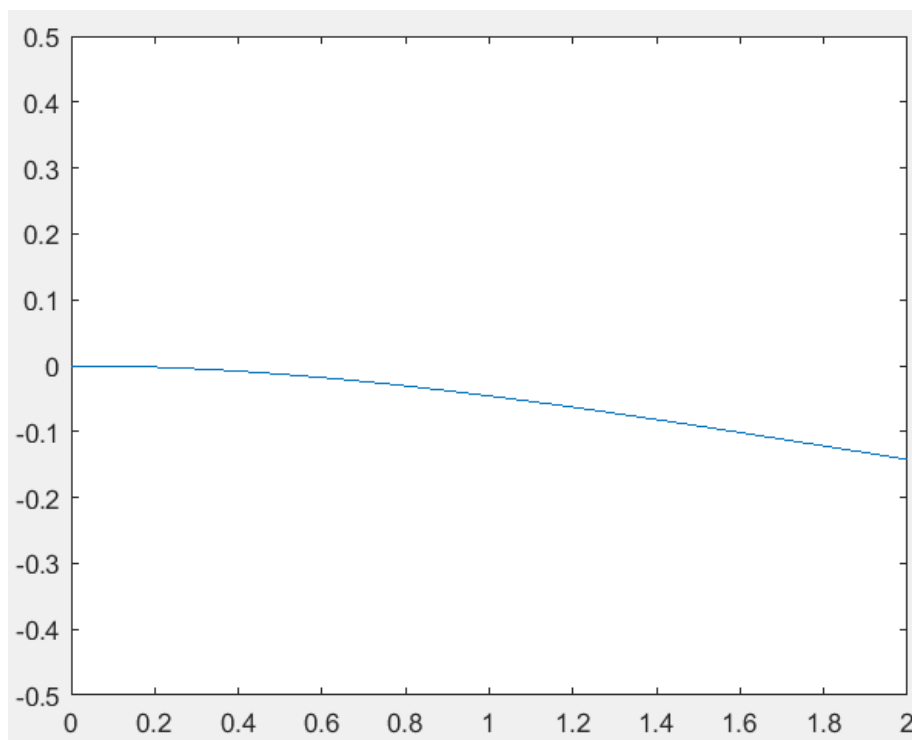
Fra grafen i oppgave c) og d) så man at den minste feilen får man når $n = 1280$. Det betyr at for å få den mest nøyaktige tilnærmingen mulig, så bør man dele bjelken i 1280 deler og bruke numerisk metode. Grunnen til at dette tallet gir minst feil er en kombinasjon av et lavt kondisjonstall og en relativt lav teoretisk feil, som til sammen gjør at $n = 1280$ vil gi den minste faktiske feilen.

4.7 Oppgave 7

For å minimere feilen er det best å bruke $n=1280$, som beregnet i oppgave 6. For å representere at en person er stå på enden av stupebrettet tenker vi at personen er plassert på de siste 30 centimeterene av stupebrettet, og at vekten til person er jevnt fordelt. Tyngden som virker fra stupebrettet blir da følgende: (w , d = bredde og tykkelse på stupebrettet)

$$f(x) = D \cdot w \cdot d \cdot g + \begin{cases} -g \frac{50}{0.3} \text{ kgm}^{-1}, & L - 0.3m \leq x \leq L \\ 0 \text{ Nm}^{-1}, & 0m \leq x < L - 0.3m \end{cases}$$

Med passende $f(x)$ definert, og strukturmatrise med $n=1280$ setter vi inn alt i likning 2.34 fra boka og løser likningsettet. Resultatet blir høyden beregnet for 1280 ulike distanser fra den festede delen av stupebrettet, se figur under.



Den ytterste delen av stupebrettet finner sted helt til høyre, den har $y = -0.1423$. Stupebrettet er altså bøyd ned litt over 14cm ved enden.

5 Konklusjon

I prosjektet har vi jobbet mye med å studere Euler-Bernoulli bjelken, som er en matematisk modell som forteller oss hvordan et materiale bøyes avhengig av hvilken last som benyttes. Oppgavene vi har løst har hatt mye å gjøre med beregninger på et stupebrett og hvilke feilmarginer som dukker opp i beregningene.

Vi startet rapporten med å bevise at 2. ordens formelen for den fjerdederiverte er gitt ved

$$f^{(iv)}(x) = \frac{f(x-2h) - 4f(x-h) + 6f(x) - 4f(x+h) + f(x+2h)}{h^4} + O(h^2)$$

Her ser vi at $f^{(iv)}(x)$ er en god tilnærming for $y''''(x)$ som blir brukt i Euler-Bernoulli modellen, siden $O(h^2) \rightarrow 0$. Men som beskrevet i delkapittel 3.7, så vil denne tilnærmingen være udefinert for x_1 . I delkapittel 4.1.2 så beviste vi en annen tilnærming $f^{(iv)}(x)$ som oppfylte randbetingelsene, og som også er definert $\forall x_i$ i uttrykket

$$\frac{f(x_i)h^4}{EI}$$

Denne approksimasjonen viser seg å være

$$y^{(iv)}(x) = \frac{16f(x+h) - 9f(x+2h) + \frac{8}{3}f(x+3h) - \frac{1}{4}f(x+h)}{h^4}$$

Disse approksimasjonene legger grunnlaget for hvordan vi programmerte strukturmatriksen A (se MATLAB-koden i delkapittel 7.1).

Med tanke på Euler-Bernoulli-teoremet har vi brukt strukturmatriksen A , som er en tilnærming av den fjerdederiverte av forskyvningen av stupebrettet. For å estimere forskyvningen på n antall punkter på stupebrettet blir strukturmatriksen brukt med størrelse $n \times n$ og setter opp en ligning for å finne forskyvningen på de n punktene. Matriksen som blir funnet i delkapittel 4.3 viser at den estimerte forskyvningen blir større jo lenger ut man kommer i stupebrettet, noe som stemmer godt med randbetingelsen gitt i delkapittel 3.10.

Videre har vi vist at den numeriske metoden for den fjerdederiverte $\frac{1}{h^4}Ay$ er eksakt i oppgave 4, og sammenlignet med f/EI er den numerisk fjerdederiverte nesten eksakt lik. Vi finner så kondisjonstallet til strukturmatriksen og feilforstørringen til tilnærmingen og ser at den er mye mindre enn kondisjonstallet som betyr at tilnærmingen er meget god.

Når det gjelder strukturmatriksen har vi vist at avrundingsfeil blir større og større jo høyere n er, altså at sjansen for feil ved avrunding blir høyere når vi deler opp stupebrettet i mindre deler. I oppgave 5 ser vi på en bjelke der den eneste massen som påvirker bjelken er dens egenmasse. Når vi da løser likningen

$Ay = b$ for $n = 10, 40, \dots, 10 \cdot 2^{11}$, så ser vi at feilen er monotont voksende etter hvert som n blir større. Siden vi kun opererer med egenmassen her, så skyldes den økende feilen det økende kondisjonstallet. Det økende kondisjonstallet fører til at sannsynligheten for avrundingsfeil øker, noe som gjenspeiles i den økende feilen.

Ved approksimering av forskyvningen med en sinusformet haug på stupebrettet ser vi at feilen i approksimeringen blir mindre og mindre med høyere n , men slutter å bli bedre etter $n = 1280$. I motsetning til oppgave 5, så bruker vi en ikke-konstant $s(x)$ her, nemlig

$$s(x) = -pg \sin\left(\frac{\pi}{L}x\right)$$

Grunnen til at feilen synker i starten skyldes at vi adderer $s(x)$ til $f(x)$ slik at vi ender opp med

$$f(x) = -480wdg + s(x).$$

Siden $s(x)$ avhenger av n , så vil tilnærmingen bli bedre jo større n blir. Men det kommer et punkt hvor kondisjonstallet blir så stort at feilen begynner å øke igjen. Maskinen klarer ikke å runde av riktig nok, og feilen øker.

I oppgave 7 ser vi hvordan stupebrettet bøyer seg når en person med masse 50 kg står på de siste 30 centimeterne av stupebrettet. Vi kommer frem til at stupebrettet bøyer seg litt over 14 cm. Her har vi brukt $n = 1280$, siden det var den verdien som ga det beste estimatet fra oppgave 6. Det at stupebrettet bøyer seg stemmer overens med randbetingelsen om at bjelken er festet i den ene enden.

5.1 Videre arbeid

For å få en mer presis modell, så kan man se på Timoshenko-Bjelke-teoremet². Denne modellen er en generalisering av Euler-Bernoulli-teoremet, og tar også hensyn til hvordan rotasjonsbøyning og deformasjon påvirker bøyningen av bjelken. Dette blir en mer fysisk korrekt modell med tanke på de forenklingene man fjerner fra Euler-Bernoulli modellen.

Videre kunne vi også ha brukt et annet programmeringspråk eller bibliotek som tillater mer nøyaktig presisjon. Med dette kan man oppnå bedre resultater når det kommer til Euler-Bernoulli modellen. For eksempel kunne man ha brukt *long double* istedenfor *double* som vi har gjort her, eller benyttet seg av *mpmath*-biblioteket i Python³.

6 Referanseliste

¹ Sauer, Numerical Analysis 2nd edition (George Mason University: Pearson Education, 2012)

² Timoshenko-bjelke-teoremet, https://en.wikipedia.org/wiki/Timoshenko_beam_theory

³ mpmath-biblioteket, <http://mpmath.org/>

7 Vedlegg

7.1 Oppgave 2

```
1 % Oppgave 2
2 function A = lagA(n)
3     e = ones(n,1);
4     A = spdiags([e, -4*e, 6*e, -4*e, e], [-2, -1, 0, 1, 2], n, n);
5
6     A(1, 1:4) = [16, -9, 8/3, -1/4];
7     A(n-1, n-3:n) = [16/17, -60/17, 72/17, -28/17];
8     A(n, n-3:n) = [-12/17, 96/17, -156/17, 72/17];
9 end
```

7.2 Oppgave 3

```
1 % Oppgave 3
2 function y = egenMasseForskyvning()
3     n = 10;
4     A = lagA(n);
5
6     L = 2;
7     d = 0.03;
8     w = 0.3;
9     tetthet = 480;
10    g = 9.81;
11    E = 1.3e10;
12
13    h = L/n;
14    I = (w*d^3)/12;
15
16    func = @(x) (-tetthet*w*d*g);
17    b = h^4 / (E*I) * arrayfun(func, h:h:L)';
18
19    y = A\b;
20
21    plot(0:h:L, [0 y']); axis equal;
22 end
```

7.3 Oppgave 4

```
1 import math
2
3 def y(x):
4     return (-42.3792/(24*(1.3*10**10)* ...
5             6.75*(10**7)))*(x**2)*((x**2)-(4*2*x)+(6*(2**2)))
6
7 def printMatrix(array):
8     for i in array:
9         print(i)
10
11 #calculating 1/h^4 * A * ye
12 def calculate(A, y):
13     result = []
14     for i in range(0, len(A)):
15         for j in range(0, len(A[i])):
16             A[i][j]*=(1/(0.2**4))
17             #print(A[i][j])
18
19     for i in range(0, len(A)):
20         sum = 0
21         for j in range(0, len(A[i])):
22             #print(y[i])
23             sum+= A[i][j]*y[j]
24         result.append(sum)
25     return result
26
27
28
29 ye = []
30
31 for i in range(1, 11):
32     ye.append(y(i*0.2))
33     print(y(i*0.2))
34
35 A = []
36
37 #A-matrisen
38 A.append([16.0,-9.0,8.0/3.0,-1.0/4.0,0,0,0,0,0,0])
39 A.append([-4,6,-4,1,0,0,0,0,0,0])
40 A.append([1,-4,6,-4,1,0,0,0,0,0])
41 A.append([0,1,-4,6,-4,1,0,0,0,0])
42 A.append([0,0,1,-4,6,-4,1,0,0,0])
43 A.append([0,0,0,1,-4,6,-4,1,0,0])
44 A.append([0,0,0,0,1,-4,6,-4,1,0])
45 A.append([0,0,0,0,0,1,-4,6,-4,1])
46 A.append([0,0,0,0,0,0,16/17,-60/17,72/17,-28/17])
47 A.append([0,0,0,0,0,0,-12/17,96/17,-156/17,72/17])
48
49
50 #printMatrix(calculate(A, ye))
51
52 Aye = calculate(A, ye)
53 res = []
```

```

54
55 #f/EI
56 fel = -42.3792/(1.3*(10**10)*6.75*(10**-7))
57
58 #lager vektor for differanse mellom f/EL * [1, 1, ... 1]^T og ...
   1/h^4 Aye
59 for i in range(0, 10):
60     res.append(abs(fel - Aye[i]))
61
62 for i in res:
63     print(i)
64
65 print(res[4]/fel)
66 print(abs(res[4]/fel)/(2**-52))
67
68
69 #printMatrix(A)

```


7.4 Oppgave 5

```
1 % Oppgave 5
2 function t = tabell()
3     format longG;
4     n_s = 10*2.^(1:11);
5
6     L = 2;
7     d = 0.03;
8     w = 0.3;
9     tetthet = 480;
10    g = 9.81;
11    E = 1.3e10;
12
13    I = (w*d^3)/12;
14
15    func = @(x) (-tetthet*w*d*g);
16    y = @(x) (func(x)/(24*E*I))*(x^(2))*(x^2-4*L*x+6*L^2);
17
18    feil = zeros(1, length(n_s));
19    kond = zeros(1, length(n_s));
20    riktig = y(L);
21
22    index = 1;
23    for n = n_s
24        A = lagA(n);
25        h = L/n;
26        b = h^4 / (E*I) * arrayfun(func, h:h:L)';
27        y = A\b;
28        feil(index) = y(end) - riktig;
29        kond(index) = condest(A);
30
31        index = index + 1;
32    end
33
34    t = table(n_s', feil', kond');
35    t.Properties.VariableNames = {'n' 'Feil' 'Kondisjonstall'};
36 end
```

7.5 Oppgave 6

```
1 function t = tabell6()
2     n_s = 10*2.^(1:11);
3
4     L = 2;
5     d = 0.03;
6     n = 10;
7     p = 100;
8     w = 0.3;
9     tetthet = 480;
10    g = 9.81;
11    n = 10;
12    E = 1.3e10;
13
14    I = (w*d^3)/12;
15
16    func = @(x) ((-tetthet*w*d*g) -p*g*sin((pi/L)*x));
17    y = @(x) func(0)/(24*E*I) * x^2 *(x^2 - 4*L*x + 6*L^2) - ...
        (g*p*L)/(E*I*pi) * ((L^3/pi^3)*sin(pi*x/L) - x^3/6 + ...
        L*x^2/2 - L^2*x/pi^2);
18
19    feil = zeros(1, length(n_s));
20    kond = zeros(1, length(n_s));
21    riktig = y(L);
22    last = zeros(0, length(n_s));
23
24    index = 1;
25    for n = n_s
26        A = lagA(n);
27        h = L/n;
28        b = h^4 / (E*I) * arrayfun(func, h:h:L)';
29        y = A\b;
30        feil(index) = abs(y(end) - riktig);
31        kond(index) = condest(A);
32        last(index) = y(end);
33        index = index + 1;
34    end
35    t = table(n_s', feil', kond');
36    t.Properties.VariableNames = {'n' 'Feil' 'Kondisjonstall'};
37
38    zoom yon;
39    plot(last);
40    set(gca, "xTick", 1:11)
41    set(gca, "xTickLabel", n_s)
42    grid;
43    hold on;
44    plot(riktig.*ones(length(last), 1));
45    hold off;
46    end
```

7.6 Oppgave 7

```
1 function res = oppg7_vekt(pos)
2     res = 0; % i utgangspunktet ikke noe ekstra vekt p brettet
3     if (pos >= 1.7) % kun siste 30cm har vekten av stuperen p seg
4         res = -9.81*50/0.3;
5     end
6 end
```

```
1 oppg7func();
2 function res = oppg7func()
3     n = 1280; % verdien som viste seg gi best tilnærming i oppg6
4
5     L = 2; % lengde p stupebrett i meter
6     d = 0.03; % tykkelse p stupebrettet
7     w = 0.3; % bredden p stupebrettet
8
9     tetthet = 480; %kg/m^3
10    g = 9.81; % tyngdefeltstyrken p jorda
11    E = 1.3e10; % materialkonstanten (youngmodulusen)
12    I = (w*d^3)/12; % arealmoment
13
14    func = @(x) ((-tetthet*w*d*g) + oppg7_vekt(x));
15
16    A = lagA(n);
17    h = L/n;
18    b = h^4 / (E*I) * arrayfun(func, h:h:L)';
19    y = A\b;
20    res = y(end);
21    disp(res)
22
23    axis([0 2 -1 0])
24
25    plot(h:h:L, y)
26    ylim([-0.5 0.5])
27 end
```