

TMR4160 prosjektrapport

Tor Erik Larsen

27. april 2015

Definering av oppgaven

A shipping company consider two different barges for a given task. The barge must be cheap and easy to build. The shipping company have therefore decided that the barge should be shaped like a shoe box. Two alternative barges are considered. :

- Alternative 1 : $L=100$, $B=10$, $D=5$
- Alternative 2 : $L=100$, $B=14$, $D=3.5714286$

Your job is here to write a program which will make it possible to visualize the motion of the barge in an irregular sea state. In addition you should do some statistical calculations. The shipping company want to operate these barges in the Northern Sea.

Since they will operate in the Northern Sea, the sea state is described by the Jonswap spectrum, with parameters $H_s=12$, $T_p=10$, $\Gamma=3$. You should calculate :

- Null krysningsperiode ($T_{m0z} = 2\pi\sqrt{M_0/M_2}$)
- Midlerer periode ($T_{m0e} = 2\pi\sqrt{M_0/M_1}$)
- Midlerer periode mellom bølge toppene ($T_{m24} = 2\pi\sqrt{M_2/M_4}$)
- Høyeste bølge på en 5 timers periode. ($H_{max}=H_{m0}\sqrt{\ln(N)/2}$, $N=\text{varighet av sjøtilstanden} / T_z$)

In the final report, you should address the above mentioned tasks. You should report the calculated transfer functions in heave and pitch. For those in the shipping company which is not used to interpret transfer functions, a visualization would be useful.

The first part of the project - where you calculate the transfer functions should be written in Fortran. The fortran program should end in writing a file containing the transfer functions. Another program, written in C should read this file, make time realizations and write the result to files which is read by GLView.

1 Forord

Denne oppgaven har blitt til med hjelp av mange timer på datasalen på Tyholt. Lærekurven har vært bratt, men samtidig har jeg lært veldig mye om programmering med lavenivåspråk. I tillegg har teorien vært koblet veldig godt opp mot tidligere hydrofag, og ikke minst TMR4182 Marin Dynamikk.

Det er ikke tatt lette valg for å komme i mål. Som Holm har formidlet, har jeg benyttet sjansen til å lære å koble opp til en terminalserver, og programmere og kompilere i et Linux/ emacs-miljø. Totalt har man lært seg både Linux, emacs, Fortran og C. Underveis i oppgaven har det også blitt mange gode diskusjoner med studiekamerater som har valgt samme oppgave.

Faget har også innebært mye modning programmeringsmessig. Mange av valgene som ble tatt tidlig i programdesignfasen fremstår nå som i beste fall uryddige. Om noe av hensikten med emnet var å innse lærekurven har dette vært svært effektivt.

Alt i alt har dette vært et meget interessant og relevant fag som i stor grad har utfordret selvstendig innsats. Takk til Håvard Holm og hans team for god veiledning og støtte underveis.

2 Sammendrag

I denne rapporten presenteres arbeidet som er gjort i emnet TMR4160 datametoder vårsemesteret 2015. Den valgte oppgaven er lektere i irregulær sjø. Et program ble skrevet i Fortran, og produserte en resultatfil med transferfunksjoner. Et program ble skrevet i C, og behandler transferfunksjonene og produserer en fil som GLView kan kjøre for å visualisere hvordan lekterne beveger seg i sjøen. Det er valgt å bruke 100 bølgekomponenter i visualiseringsprogrammet, plukket over hele frekvensspekteret. Grunnen til at ikke hele frekvensspekteret er brukt er å minke beregningstid, og ikke minst ta litt mindre energi fra spekteret i den grove sjøtilstanden. Det er også kjørt en resultatkontroll i C-programmet, hovedsaklig ved å plote respons mot en bølgekomponent som i regulær sjø. Transferfunksjonene viser en forskjell i utslag i lekterne, som favoriserer den bredeste lekteren. Dette kan skyldes blant annet at den bredeste lekter B har dobbelt så stor verdi for dempekoeffisientene. Likevel viser animasjonen at forskjellen kanskje ikke vil være så stor i virkeligheten. Rederiet frarådes å gå videre med valgt design. Fribordet bør økes, og det bør vurderes å montere overbygg og annen miljøbeskyttelse.

Innhold

1 Forord	iii
2 Sammendrag	iv
3 Innledning	1
4 Teori	1
4.1 Antagelser og forenklinger	1
4.2 Statistisk modellering av bølger	2
4.3 Overføringsfunksjoner	5
4.4 Strukturens respons	13
5 Om beregningsprogrammet	14
5.1 Målsetting med program	14
5.2 Programstruktur	14
6 Resultater fra programmet	17
6.1 Bølgestatistikk	17
6.2 Beregnede transferfunksjoner for hiv og stamp	17
6.3 Respons, havflate og visualisering	18
6.4 Andre resultater fra programmet	20
7 Diskusjon	23
7.1 Konsekvens av forenklinger og antagelser	23
7.2 Bølgespekter og feilkilder	23
7.3 Lekterrespons og feilkilder	23
7.4 Generalisering av kode	23
8 Konklusjon og videre arbeid	23
A Programkode	25
A.1 Bruksanvisning for DMET DELUXE 2000	25
A.2 Vedlagt kode	25

3 Innledning

Når et rederi skal investere i nye lektere, kan det være en stor fordel å analysere de ulike alternativene før man setter i gang bygging. Matematiske modeller kan programmeres til å kjøre på en datamaskin, og i stor grad gi en god pekepinn på om de aktuelle alternativene passer inn i tiltenkt havmiljø.

Denne oppgaven presenterer to ulike lektere som er tiltenkt å plasseres i Nordsjøen. Ved å beregne en bølgeoverflate ut fra velkjente bølgespektra for dette området kan det produseres en rimelig god animasjon av hvordan disse vil takle en relativt grov sjøtilstand .

To programmeringsspråk er benyttet: FORTRAN (“vitenskapelig C”) og C. Fortran-delen har beregnet transferfunksjoner og bølgestatistikk. C-programmet har beregnet respons, bølgebevegelser og lagt dette inn i en animasjonsfil.

Teorien er drøftet med tanke på forståelse og et perspektiv foran ren reproduksjon av matematiske formler. Det er så langt mulig lagt opp til at rapporten skal være skrevet på norsk, kodespråk unntatt.

Rapporten er skrevet for at rederiets representanter skal kunne lese og forstå innholdet, samt kjøre program og visualisering. I tillegg er det lagt vekt på at tilstrekkelig teknisk språk er brukt for at fagpersoner skal vurdere kvaliteten i teori og programmering.

4 Teori

I denne inndelingen vil det diskuteres teori. I stedet for å reprodusere formler er det lagt vekt på å forklare ut fra forfatterens forståelse. Den samlede teorien er omfangsrik, men det forsøkes å begrense det presenterte utvalget.

4.1 Antagelser og forenklinger

Det antas i denne inndeling at leseren har kjennskap til grunnleggende marinteknisk teori. Vanskelige begreper forsøkes forenkles, men denne inndelingen er primært rettet mot leseren som ønsker å ettergå det teoretiske fundamentet som rapporten bygger på.

Det antas i oppgaven at bølgene består av en rekke langkammede bølger som kommer fra en eneste retning θ . Dermed kan bølgesituasjonen betraktes som todimensjonal (Myrhaug og Lian 2009, side 3). Dette er en betydelig forenkling, som medfører at vi kan la lekteren ta imot sjøen rett mot baug. Vi slipper dermed å forholde oss til sjø som kommer fra alle retninger samtidig, slik det av og til kan virke som ute på havet. Vi kan dermed tenke oss at lekteren ligger langs en langside i et havbasseng, med uregelmessige bølger som er produsert fra en bølgemaskin i en av kortendene.

Det antas at lekteren ligger på såkalt dypt vann, som betyr at vanndybden h er så stor at vannpartiklene som beveges av bølgeenergien ikke virker ned til havbunnen. Dermed slipper man også å ta hensyn til at havbunnen påvirker bølgene. En tommelfingerregel er at man har dypt vann når h er større enn halve bølgelengden λ (les Pettersen 2007, side 1.32). Denne tommelfingerrelen stammer muligens igjen fra det faktum at det dynamiske trykket $-\rho \cdot \frac{\partial \phi}{\partial t}$ nede i halve bølgelengdens dyp er kun 4% av trykket oppe på vannoverflaten $z = 0$ (Faltinsen 1990, side 21). Uansett er dypvannsantagelsen en betydelig forenkling, siden det eneste som vil påvirke bølgehastigheten er bølgelengden ut fra dispersasjonsrelasjonen $\omega^2 = kg$

Det er ikke gitt av oppgaveteksten, men det antas at lekterne skal ligge i ro, og at strøm ikke er av gjeldende styrke. I så fall tilfredsstilles betingelsene for å kunne regne med lineær bølgeteori. Det betyr igjen at bølgeamplituden er proporsjonal med hastighetspotensialet, og vil være små i forhold til bølgelengde og legemets størrelse (fra Faltinsen 1990, side 15).

En viktig antagelse når det gjelder lekterens bevegelser i vannet, er at man ser bort fra det som kalles viskøs strømming. I stedet for å ta hensyn til at lekterens flater vil danne friksjon med vannet, legges det inn en empirisk beregnet tilleggsmasse A_{33} i hiv. For stampebevegelsen kan man også definere tilleggsmasse A_{55} på samme måte, men gitt symmetri om langskips tyngdepunkt, vil disse kansellere hverandre for og akter. Ved bruk av denne antagelsen benyttes det som kalles potensialteori, og vil for en så enkel geometri som lekteren utgjør være en anerkjent tilnærming til virkelige resultater. I potensialteorien vil skjærkraften i væsken være 0.

4.2 Statistisk modellering av bølger

For å modellere det som skjer i virkelige bølger, trenger man faste definisjoner som man kan forholde seg til hver eneste gang man regner. Alment aksepterte definisjoner gir stabile resultater uavhengig av hvem som gjør beregningene, og medfører at strukturer som skal gå i sjøen er antatt klar for belastningene selv på papiret.

Selvsagt innebærer slike definisjoner at man går litt bort fra virkeligheten, ofte i form av forenkende antagelser. Likevel er anerkjente antagelser vil mest sannsynlig gi feil som både er alment kjente og relativt små i forhold til sluttresultatet. Et eksempel på en slik forenkling er begrepet sjøtilstand. For noen som er interesserte i å beregne på bølgestatistikk kan sjøtilstanden løst defineres som en gjennomsnittlig bølgehøyde H_s , også kalt signifikant bølgehøyde over en periode mellom 20 minutter og 10 timer.

En sjøgang består av en rekke regulære bølger, hver med sin egen fasevinkel og frekvens. Disse bølgene utgjør overflatehevingen i en sjøgang slik:

$$\zeta = \sum_{j=1}^N A_j \sin(\omega_j t - k_j x + \epsilon_j) \quad (1)$$

Ligning 1 (fra Faltinsen 1990, ligning 2.22) består av overflateheving ζ , bølgens amplitude A_j , frekvens ω , bølgetall k_j og fasevinkel ϵ_j . Det vises til dispersjonsrelasjonen for sammenhengen mellom ω og k for vårt dype vann. Det er viktig å merke seg at overflatehevingen fra ligning 1 er momentanfordelt Gauss, som vil være naturlig i og med at bølgene har en positiv og negativ amplitude som svinger rundt $\zeta = 0$.

Om man ønsker å simulere en sjøgang på en gitt lokasjon over tid, bruker man ligning 1 og legger inn et stort antall tilfeldige frekvenser og faseforskyvninger. Faltinsen anbefaler antall bølgekomponenter $N = 1000$.

Ligning 1 angir hvordan man kan beregne overflatehevingen for en lokasjon. For å analysere en sjøgang ytterligere, er det laget en kobling mellom bølgens amplitude og energiinnhold. Et standardisert bølgespekter er:

$$\frac{1}{2} A_j^2 = S(\omega_j) \Delta \omega \quad (2)$$

Her er amplituden A_n , funksjonsverdien $S(\omega)$ og det konstante intervallet mellom frekvensene $\Delta \omega$. Spekteret er definert slik at et areal vil utgjøre energimengden i intervallet $S(\omega) \Delta \omega$

Standardiserte bølgespektra

Vi har nå drøftet overflateheving av sjø, og koblingen mellom amplitude og tanken bak en funksjon som viser en sjøgangs generelle energinivå S som en funksjon av frekvensen ω . Neste steg er å finne spektra som mariningeniører kan bruke når man designer fartøy som flyte kan. Denne rapporten skal ikke utlede hvordan standardiserte spektra dannes, men om man har målt overflatehevingen på et område, kan man benytte sammenhengen mellom miljøpåvirkning som vind, og bølgenes frekvens kobla mot tyngdeakselerasjonen.

$$S(\omega) = \frac{g^2}{\omega^5} * f\left(\frac{V\omega}{g}\right) \quad (3)$$

Her er g tyngdefeltsakselerasjon, bølgefrequens ω og vindhastighet V . Det er klart fra formelen at energien vil avtar med ω^{-5} med økende bølgefrequens. Enheter er det samme som bølgespekteret [$m^2 * s$] Vi skal se eksempel på dette. Det er naturlig at vindhastighet er eneste parameter i spekteret, siden det er vinden som driver bølgene over tid. Skjærkraften som oppstår ved at vinden blåser over vannet, og medfølgende hastighetsprofil, gjør at vannet får en horisontal hastighet.

Som Philips angivelig beskrev det, vil høye frekvenser ha minimal påvirkning på det totale energiinnholdet i spekteret (Myrhaug og Lian 2009, side 8). Som vi husker, er det totale energinivået arealet under hele spektergrafen.

Et av de mest brukte spektra er PM-type-spektre. Disse kan formuleres slik:

$$S(\omega) = \frac{A}{\omega^5} e^{-\frac{B}{\omega^4}} \quad (4)$$

PM-spekterets form kan gjenkjennes fra ligning 3, men der A inneholder tyngdeakselerasjonen, og B inneholder forholdet mellom tyngdeakselerasjon og vindhastigheten.

I oppgaven er vi bedt om å bruke et spesifikt spekter for Nordsjøen som kalles JONSWAP, og er laget for å kunne forutsi bølgeforholdene i Nordsjøen. En forsøksvis enkel forklaring på hva som skiller et ordinært PM-spekter og JONSWAP-spekteret er først og fremst antall parametere. JONSWAP har 5 parametere $\alpha, \gamma, \sigma_a, \sigma_b$ i motsetning til PM sitt ene. Det som er verdt å merke seg med disse 5 parameterene er at bortsett fra empiriske verdier finnes vindhastighet V og vindfang x . Rent fysisk er det ingenting som er endret i forhold til et PM-spekter, det er fortsatt vind som driver bølgene. Det som er annerledes er at spekteret ikke er basert på at vinden har fått drive bølgene til de er fullt utviklede. Om det er relevant kan man legge inn vindfanglengden i meter for det aktuelle området.

I tillegg til antall parametere, er det en annen vesentlig forskjell på spekterdefinisjonene. Energifordelingen langs ω -aksen er ulik, spesifikt definisjonen av maksimalt energinivå. I JONSWAP er energien mer konsentrert rundt ω_p , noe som kanskje reflekterer bedre forholdene i Nordsjøen. Matematisk er denne energifordelingsulikheten beskrevet som parameteret γ :

$$\gamma = \frac{S_{JONSWAP, max}}{S_{PM, max}} \quad (5)$$

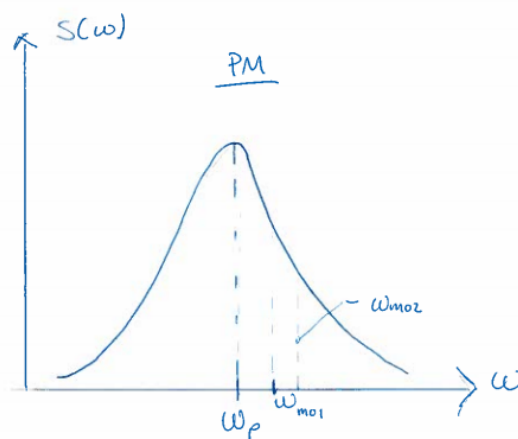
Enkelt forklart kan γ multipliseres med PM-spekteret, men det finnes ulike løsninger på dette. Uansett er spekteret for oppgaven gitt i oppgaveteksten, og var åpenbart for hårreisende til at Latex godtok det.

Sammenhengen mellom bølgeparametere for PM

I oppgaven bes det om 4 ulike statistiske verdier beregnes og presenteres. Mens de statistiske beregningene utredes mer nøye i neste avsnitt, skal vi nå se nærmere på disse parameterne:

- Signifikant bølgehøyde H_{m0}
- Toppfrekvens ω_p
- Topperiode T_p
- Spektermomentene m_0, m_1 og m_2
- Middelperiodene T_{m01} og T_{m02}

Den signifikante bølgehøyden H_{m0} eller H_s er definert som gjennomsnittet av de 1/3 høyeste bølgene over en gitt tidsperiode. Denne tidsperioden er ofte mellom 20 minutter og 10 timer. En sjøtilstand er i bølgestatistikken karakterisert med en konstant H_s for en gitt varighet. Denne forenklingen medfører for eksempel at man kan produsere tabeller for langtids bølgestatistikk.



Figur 1: Skisse av PM-spekter

Som vist på figur 1 kan man se at det finnes en frekvens ω_p som innehar den høyeste energimengden i spekteret. Man bør være obs på toppfrekvensen fordi den bør ligge komfortabelt unna egenfrekvensen til strukturen man konstruerer, for å unngå sannsynligheten for resonans. Topperioden henger sammen med denne gjennom $T_p = 2 * \pi / \omega_p$.

For å diskutere de gjenstående frekvensene er det nyttig å presentere uttrykket for momenter:

$$m_n = \int_0^{\infty} \omega^n S(\omega) d\omega \quad (6)$$

Vi ser at for m_0 blir uttrykket det samme som spekteruttrykket i ligning (2). Det betyr at m_0 utgjør første arealmoment av all energien bølgespekteret inneholder. Videre er middelperioden definert som $\omega_{01} = m_1/m_0$. Rent fysisk er dette en vektet momentsum som er delt på et totalmoment, og vi får et gjennomsnitt av bølgeenergi på frekvensen som vist på figur 1. Når det gjelder m_2 er det svært interessant at definisjonen gir oss annet arealmoment ut fra matematikken.

Middelperiodene T_{m01} og T_{m02} følger naturlig av foregående diskusjon.

Korttids bølgestatistikk

Det vil her gjøres svært kort greie for resultatene som er påkrevd i oppgaven. All teori er hentet fra (Myrhaug og Lian 2009, kapittel 3)

Midlere periode

$$T_1 = T_{m01} = 2\pi \frac{m_0}{m_1}$$

er definert som tyngdepunktet i spekteret. Nullkryssperioden

$$T_2 = T_{m02} = 2\pi \sqrt{\frac{m_0}{m_2}}$$

er definert som snittperioden mellom hver gang en bølge krysser nullaksen. Midlere periode mellom bølgetoppene

$$T_{24} = 2\pi \sqrt{\frac{m_2}{m_4}}$$

er definert som nettopp snittperioden mellom toppene, ikke nullkrysningene. Til sist er maksimal bølgehøyde for fem timers varighet beregnet som

$$H_{max} = H_s \sqrt{\frac{\ln N}{2}}$$

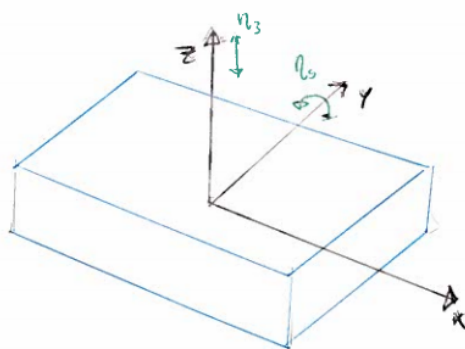
som er et statistisk estimat for hvor store bølger man kan møte i en gitt sjøtilstand H_s .

Med bølgestatistikken på plass er det på tide å diskutere hvordan lekterene skal oppføre seg.

4.3 Overføringsfunksjoner

Som det kanskje ligger i ordet, skal en overføringsfunksjon sørge for at energien som bølgene representerer resulterer i en kraft som flytter et legeme en viss distanse. Når energien kan uttrykkes i form av bølgens utslag, og bølgene har en viss regelmessighet, er det ikke langt til å konkludere med at lekteren i vårt tilfelle vil bevege seg med en viss periodisk tilnærming. I regulære bølger vil lekteren over tid oppnå en svingebevegelse som er et direkte og periodisk svar på energien fra bølgene, i form av hiv og stamp.

Transferfunksjoner betyr en oppskrift på hvordan man kan forutsi ent legemes bevegelser, gitt at man antar en matematisk modell av en sjøtilstand. Man vet hvordan bølgene oppfører seg, dermed kan man beregne hvordan energien i bølgene fører til bevegelse i lekteren.



Figur 2: Illustrasjon av lekterens frihetsgrader

Figur 2 viser de to frihetsgradene som skal presenteres i oppgaven. Lekteren kan selvsagt bevege seg i alle seks frihetsgrader, men som en forenkling er det stamp og hiv som skal analyseres i denne presentasjonen.

Stamp og hiv beregnes hver for seg, men som det skal vises er bevegelsene ikke uavhengig av hverandre. Når lekteren dykker ned med baugen samtidig som hekken løftes, vil det gi et visst bidrag til en ren hivbevegelse. Også den andre veien rundt.

Motivasjonen for å finne overførings-funksjonene til lekteren, er at man kan sette dem inn i en responsfunksjon. Selve overførings-funksjonen er da dimensjonsløs og plottes som en funksjon mot ω

$$|H_k(\omega_j)| = \frac{\eta_k A}{\zeta_a} \quad (7)$$

som igjen settes inn i responsligningen

$$\sum_{i=1}^N A_j |H_k(\omega_j)| \sin(\omega_j t + \delta(\omega_j) + \epsilon_j) \quad (8)$$

der A_j er en amplitude fra en spesifikk regulær bølge, ω_j er frekvensen for den regulære bølgekomponenten, $\delta(\omega_j)$ er fasevinkelen for legemets respons og ϵ_j er den tilfeldige fasevinkelen for en bølgekomponent.

Hele poenget er at en enkelt overføringsfunksjon representerer kun en regulær bølge, mens responsen kan representere summen av alle regulære bølger som en irregulær sjøtilstand representerer (på grunn av linearitet). Om man i tillegg setter inn (2) for amplituden i ligning (8) er ringen sluttet! På mest elegante og matematisk korrekte vis er et bølgespekter koblet mot en overføringsfunksjon for å gi en respons i en tilnærmet riktig sjøtilstand.

Hydrodynamiske koeffisienter

Koeffisientene kommer fra differensialligningen. Som vist i en tidligere prosjektrapport (Bøckmann 2008) må det brukes polynomregresjon på en kurve fra marinkompendiet (fra Pettersen 2007, side 3.30) for å finne A_{33} og B_{33} som funksjoner av ω . Koeffisientene for stamp kan finnes (fra Faltinsen 1990, side 56)

$$\begin{aligned}
A_{55} &= \int_{-L/2}^{L/2} x^2 A_{33}^{2D} dx \\
&= A_{33}^{2D} \frac{L^3}{12} \\
&= A_{33} \frac{L^2}{12}
\end{aligned} \tag{9}$$

og

$$\begin{aligned}
B_{55} &= \int_{-L/2}^{L/2} x^2 B_{33}^{2D} dx \\
&= B_{33}^{2D} \frac{L^3}{12} \\
&= B_{33} \frac{L^2}{12}
\end{aligned} \tag{10}$$

De gjenopprettende faktorene representerer som sagt oppdriftens vilje til å beholde likevekt i vannet. $F_{gjenopprett} = C_{kj}\eta_j$. I hiv er dette en vertikal motkraft, mens for stamp er dette et moment med en arm mellom oppdriftssenteret og tyngdepunktet Disse er

$$C_{33} = \rho g \nabla_{lekter} \tag{11}$$

som blir massedeplasementet multiplisert med tyngdeakselerasjonen, samt

$$C_{55} = \rho g \nabla_{lekter} GM_L \tag{12}$$

$$\begin{aligned}
GM_L &= KB + BM - KG \\
&= KB + \frac{I_{55}}{\nabla_{lekter}} - KG
\end{aligned}$$

hvor lekterens vertikale tyngdepunkt G , oppdriftssenter B og respektive avstand til kjølen K er relativt enkelt å finne. Volumdeplasementet $\nabla_{lekter} = L * B * D$ og I_{55} er fra (28) som beregnes senere i rapporten.

Hivbevegelsen

Vi skal nå presentere bevegelsesligningen til hivbevegelsen. Newtons andre lov generelt er

$$M\ddot{a} = \sum F \tag{13}$$

som for vårt svingesystem i hivbevegelse gir en andreordens inhomogen differensialligning med en periodisk påtrykt kraft, av formen

$$(M + A_{33})\ddot{\eta}_3(t) + B_{33}\dot{\eta}_3(t) + C_{33}\eta_3(t) = F_3 \sin(\omega t) \quad (14)$$

der M er lekterens massedeplassment, A_{33} er den såkalte tilleggsmassen, $\ddot{\eta}_3$ er akselerasjon i vertikal retning, B_{33} er dempefaktor som følge av bølgedannelse ut fra lekter, $\dot{\eta}_3$ er vertikal hastighet, C_{33} er gjenvinnende faktor som følge av lekterens gjenopprettende oppdriftskraft, η_3 er vertikal posisjon og F_3 er den påtrykte periodiske kraften som lekteren opplever fra bølgene.

Som vi husker fra ligning (7) er det responsvariabelen η_3 vi skal ha ut fra ligning (14). Vi gjetter at løsningen skal ha formen

$$\eta_3(t) = \eta_{3A} \sin(\omega t + \epsilon) \quad (15)$$

der det vertikale utslaget η_3 er bestemt som produktet av amplitudekoeffisienten η_{3A} og svingningen $\sin(\omega t + \epsilon)$. Ved å tidsderivere (15) to ganger og sette inn i (14), får man uttrykket

$$-\omega^2 \eta_{3A} \cos(\omega t + \epsilon)(M + A_{33}) - \omega B_{33} \eta_{3A} \sin(\omega t + \epsilon) + C_{33} \eta_{3A} \cos(\omega t + \epsilon) = F_3 \cos(\omega t) \quad (16)$$

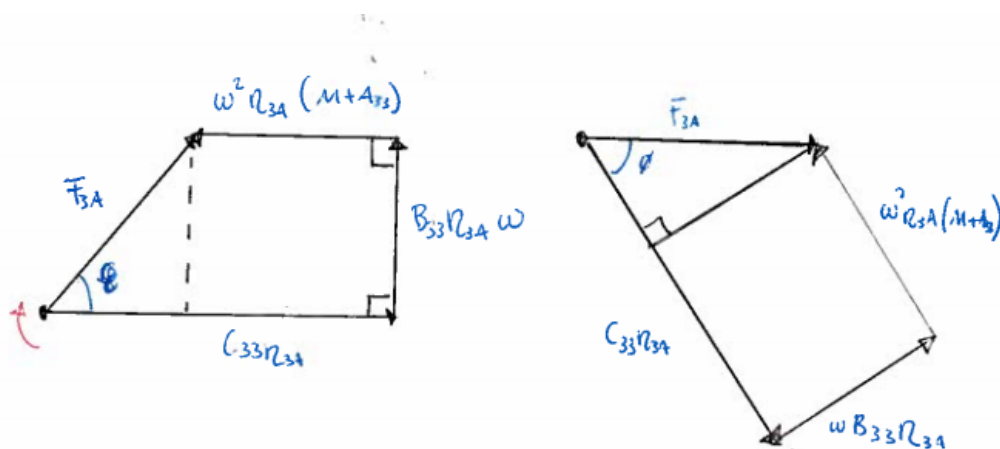
Om man forskyver akselerasjonsleddet 90 grader i forhold til hastighetsleddet, som igjen er forskyvet 90 grader i forhold til posisjonsleddet, får man (Morin 2009, side 22)

$$-\omega^2 \eta_{3A} \cos(\omega t + \epsilon + \pi)(M + A_{33}) - \omega B_{33} \eta_{3A} \sin(\omega t + \epsilon + \frac{\pi}{2}) + C_{33} \eta_{3A} \cos(\omega t + \epsilon) = F_3 \cos(\omega t) \quad (17)$$

som på sin side kan skrives som

$$\omega^2 \eta_{3A} \cos(\omega t + \epsilon)(M + A_{33}) + \omega B_{33} \eta_{3A} \cos(\omega t + \epsilon) + C_{33} \eta_{3A} \cos(\omega t + \epsilon) = F_3 \cos(\omega t) \quad (18)$$

Vi merker oss at akselerasjonsleddet har byttet fortegn, og at hastighetsleddet har blitt positiv cosinus. Vi fortsetter på utledningen av differensialligningen ved å vektorisere koeffisientene.



Figur 3: Geometrisk tolkning av (18) (Morin 2009, side 22)

Figur 3 viser den geometriske tolkningen av ligning (18), der koeffisientene er tegnet som vektorer, faseforskjøvet 90 grader i henhold til hverandre. Størrelsen på η_{3A} er valgt slik at vektorsummen er F_{3A} . Ved å rotere geometrien med den negative ϵ ser vi at summen av de horisontale vektorsommene blir $F_{3A} \cos(0)$, som på grunn av linearitet gjelder for alle vinkler.

Ved geometriske betraktninger av 3 ser vi at fasevinkelen finnes fra

$$\epsilon = \arctan(\epsilon) = \arctan\left(-\frac{\omega B_{33}}{C_{33} - (M + A_{33})\omega^2}\right) \quad (19)$$

hvor η_{3A} kanselleres fra teller og nevner, og negativt fortegn satt inn for hånd fra negativ fasevinkel.

Pytagoras for rettvinklet trekant med hypotenus F_{3A} gir

$$F_{3A}^2 = \eta_{3A}^2 (B_{33}\omega^2)^2 + \eta_{3A}^2 (C_{33} - (M + A_{33})\omega^2)^2 \quad (20)$$

som løses for responsvariabelen η_{3A} og innsatt i (15) gir uttrykket

$$\eta_3(t) = \eta_{3A} \sin(\omega t + \epsilon) = \frac{F_{3A}}{\sqrt{(C_{33} - (M + A_{33})\omega^2)^2 + B_{33}^2\omega^2}} \cos(\omega t + \epsilon) \quad (21)$$

$$\implies \eta_{3A} = \frac{F_{3A}}{\sqrt{(C_{33} - (M + A_{33})\omega^2)^2 + B_{33}^2\omega^2}} \quad (22)$$

som betyr at et enkelt utslag η_3 fra en enkelt bølgefrequens er en funksjon av eksitasjonskraften F_{3A} fra den enkelte bølgefrequensen.

Nå gjenstår det å finne F_{3A} , og vi løser dette ved å dele hovedproblemet opp i to delproblemer. Delproblem A tar for seg hvilken innvirkning bølgene har på en fastholdt lekter, mens delproblem B tar for seg venstre side av ligningen og man ser for seg at den trykkes opp og ned i vannet og lager bølger (damping) og møter motstand/ stivhet C.

Høyre side av ligningen består av Froude-Kriloff-krefter, som representerer det utforstyrrede trykkfelt i bølgene. Deretter kommer diffraksjonen som betyr forstyrrelse på trykkfeltet som følge av legemet i vannet. Linearitet medfører at kreftenes enkeltbidrag kan summeres. Disse kreftene er

$$F_{3e}(t) = F_{FK} + F_d \quad (23)$$

der

$$\begin{aligned} F_{FK} &= \iint_S p n_3 ds \\ &= \rho g \zeta_a e^{kz_1} B \int_{-L/2}^{L/2} \sin(\omega t - kx) dx \end{aligned}$$

der z_1 er dybde fra havoverflate ned til lekterbunnen. Bredden B ganges inn for å gjøre trykket $[N/m^2]$ om til kraft $[N/m]$. Froude-Kriloff-kreftene handler om å integrere trykket fra bølgene over undervansskroget til lekteren (Holm 2015).

Diffraksjonskraften handler om å lage et hastighetspotensiale som motvirker partikkelhastigheten i grensesjiktet til lekterskroget (Holm 2015), og skrives

$$\begin{aligned} F_d &= A_{33}^{2D} + a_3 \\ &= -\omega^2 \zeta_a A_{33}^{2D} e^{kz_2} \int_{-L/2}^{L/2} \sin(\omega t - kx) dx \end{aligned}$$

hvor A_{33}^{2D} er tilleggsmassen for hver stripe av lekterskroget i henhold til stripeteorien. I dette tilfellet er lekteren symmetrisk i både xz- og yz-planet, dermed får hver stripe identisk tilleggsmasse som summert blir A_{33} . Summen av bidragene blir

$$F_{3e}(t) = F_{3A} \sin(\omega t) = \zeta_a [\rho g e^{kz_1} B - \omega^2 A_{33}^{2D} e^{kz_2}] \frac{2}{k} \sin\left(\frac{kL}{2}\right) \cdot \sin(\omega t) \quad (24)$$

$$\implies F_{3A} = \zeta_a [\rho g e^{kz_1} B - \omega^2 A_{33}^{2D} e^{kz_2}] \frac{2}{k} \sin\left(\frac{kL}{2}\right) \quad (25)$$

hvor $k = \omega^2/g$ er bølgetall for dypt vann og L er lekterens lengde. Vi merker oss igjen at vi ikke skal ha tidsrealiseringen av kreftene, kun et utslag som følge av en regulær bølge med en spesifikk frekvens.

Lekterens symmetri gjør at hiv- og stampebevegelsen er helt ukoblet. Dermed kan man sette opp to ukoblede overføringsfunksjoner for stamp og hiv. Ligning (25) innsatt i (22) gir

$$\begin{aligned} |H_3 \omega_j| &= \left| \frac{\eta_{3A}}{\zeta_a} \right| \\ &= \left| \frac{[\rho g e^{kz_1} B - \omega^2 A_{33}^{2D} e^{kz_2}] \cdot \frac{2}{k} \cdot \sin\left(\frac{kL}{2}\right)}{\sqrt{(C_{33} - (M + A_{33})\omega^2)^2 + B_{33}^2 \omega^2}} \right| \end{aligned} \quad (26)$$

Stampebevegelsen

Bevegelsesligningen for stampebevegelsen er

$$(I_{55} + A_{55})\ddot{\eta}_5 + B_{55}\dot{\eta}_5 + C_{55}\eta_3 = \bar{F}_{5e} \sin(\omega t)$$

som løst på samme måte som (19) gir

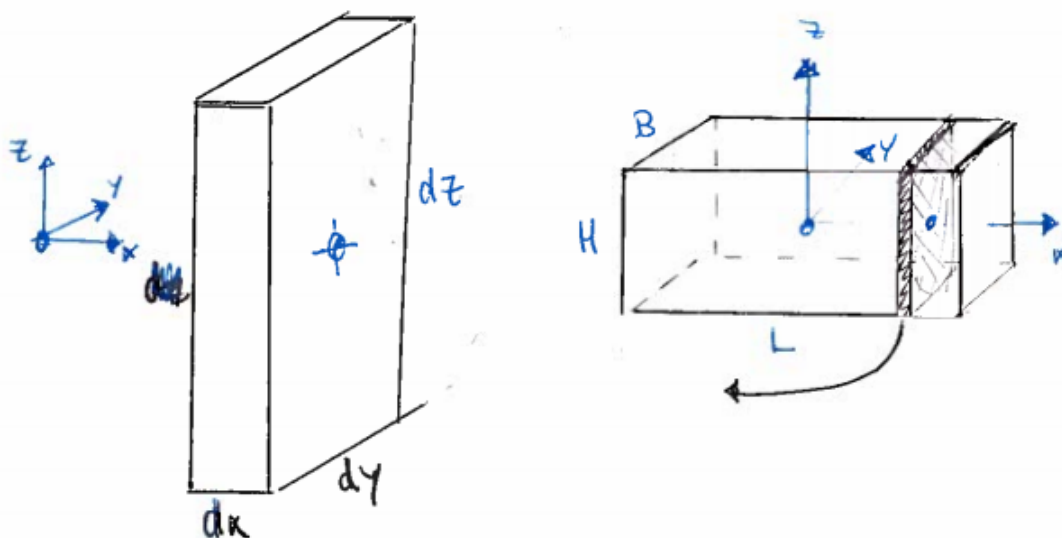
$$\eta_{5A} = \frac{F_{5A}}{\sqrt{(C_{55} - (I_{55} + A_{55})\omega^2)^2 + B_{55}^2 \omega^2}} \quad (27)$$

hvor massedeplasementet er erstattet med andre arealmoment om y-aksen; I_{55} . Massetregghetsmomentet I_{55} er beregnet ut fra (ligning (3.7) Pettersen 2007, side 3.3)

$$I_{55} = Mr_{55}^2 = \sum_i (x_i^2 + z_i^2) dm \quad (28)$$

der andre likhet består av M som er lekterens masse, og r som avstanden fra et geometrisk nullpunkt til massesenteret for hele lekteren. Den tredje likheten kan brukes til å beregne massetregghet for en assymetrisk massefordeling. Ut fra kjennskap til de ulike masseelementene kan man summere avstand fra nullpunkt til hver masse, multiplisert med massen til det enkelte elementet. Matematisk kalles dette for en vektet sum. Vi merker oss at for våre lektere strykes z^2 fra (28) på grunn av at lekterens vertikale massesenter ligger i $z = 0$.

Våre lektere har symmetri både langskips og tverrskips. Dermed kan vi tenke oss at lekteren er oppdelt i langskips skiver som vist på figur 4.



Figur 4: Skive fra leker

Vi vet lekterens masse M ut fra gitt volumdeplasement. Ut fra figur 4 ser vi også at volumet til hele lekteren er $L \cdot B \cdot H$, der $H = 2D$. Tilsvarende er volumet for en lekterskive i utgangspunktet $dV = dx \cdot dy \cdot dz$. For å finne hvor stor del av massen en skive utgjør kan vi sette opp

$$\begin{aligned} dm &= \frac{dV}{V} \cdot M \\ &= \frac{dx \cdot dy \cdot dz}{L \cdot B \cdot H} \cdot M \\ &= \frac{dx}{L} \cdot M \end{aligned}$$

Vi merker oss at ut fra geometrien i figur 4 strykes dy mot B og dz mot H . Vi setter inn i (28), med $z^2 = H^2$, og får

$$\begin{aligned} I_{55} &= \sum_i \left((x_i^2 + H^2) \cdot \frac{dx}{L} \cdot M \right) \\ &= \frac{M}{L} \cdot \sum_i \left((x_i^2 + H^2) \cdot dx \right) \end{aligned}$$

som når vi lar $dx \rightarrow 0$ gir integralet der lektergeometrisk symmetri tillater integrasjonsgrensen å gå fra 0 til L . Nevnte grenseflytting er behørig kompensert ved å multiplisere integralet med 2 og gir

$$\begin{aligned} I_{55} &= \frac{2M}{L} \int_0^{L/2} (x^2 + H^2) dx \\ &= \frac{2M}{L} \left[\frac{x^3}{3} + xH^2 \right]_0^{L/2} \\ &= \frac{2M}{L} \left[\frac{L^3}{24} + \frac{LH^2}{2} \right] \\ &= M \left[\frac{L^2}{12} + H^2 \right] \end{aligned}$$

som forhåpentligvis er matematisk og fysisk riktig.

Videre gir formelen for moment $M = \vec{\omega} \times \vec{r}$ at $F_{5e} = (zF_1 - xF_{3e})$, hvor vi neglisjerer kraften F_1 i horisontal retning, og står igjen med at $F_{5e} = -xF_{3e}$. Vi kan dermed skrive Froude-Kriloff- og diffraksjonskreftene for stamp som

$$\begin{aligned} F_{FK,5} &= \iint_S x p n_5 ds \\ &= \rho g e^{kz_1} B \int_{-L/2}^{L/2} x \sin(\omega t - kx) dx \end{aligned}$$

og

$$\begin{aligned} F_{d,5} &= -x(A_{33}^{2D} + a_3) \\ &= \omega^2 A_{33}^{2D} e^{kz_2} \int_{-L/2}^{L/2} x \sin(\omega t - kx) dx \end{aligned}$$

og ender opp med

$$F_{5e}(t) = F_{5A} \cdot \cos(\omega t) = \frac{1}{k} [\rho g e^{kz_1} B + \omega^2 A_{33}^{2D} e^{kz_2}] \cdot \left(L \cos\left(\frac{kL}{2}\right) - \frac{2}{k} \sin\left(\frac{kL}{2}\right) \right) \cdot \cos(\omega t) \quad (29)$$

$$\implies F_{5A} = \frac{1}{k} [\rho g e^{kz_1} B + \omega^2 A_{33}^{2D} e^{kz_2}] \cdot \left(L \cos\left(\frac{kL}{2}\right) - \frac{2}{k} \sin\left(\frac{kL}{2}\right) \right) \quad (30)$$

som innsatt i (27) gir stamputslaget

$$\eta_{5A} = \frac{[\rho g e^{kz_1} B + \omega^2 A_{33}^{2D} e^{kz_2}] \cdot \left(L \cos\left(\frac{kL}{2}\right) - \frac{2}{k} \sin\left(\frac{kL}{2}\right) \right)}{k \cdot \sqrt{(C_{55} - ((I_{55} + A_{55})\omega^2)^2 + B_{55}^2\omega^2)}} \quad (31)$$

som igjen innsatt i (7) gir den dimensjonsløse overføringsfunksjonen for stamp

$$\begin{aligned}
|H_5\omega_j| &= \left| \frac{\eta_5 A}{\zeta_a k} \right| \\
&= \left| \frac{[\rho g e^{kz_1} B - \omega^2 A_{33}^{2D} e^{kz_2}] \cdot (L \cos(\frac{kL}{2}) - \frac{2}{k} \sin(\frac{kL}{2}))}{k \cdot \sqrt{(C_{55} - ((I_{55} + A_{55})\omega^2)^2 + B_{55}^2 \omega^2}} \right|
\end{aligned} \tag{32}$$

og vi merker oss at bølgetallet k er satt i nevner på høyre siden for lesbarhet, for venstre side som et tillegg til bølgeamplituden ζ_a for å gjøre overføringsfunksjonen dimensjonsløs. Vi merker oss i tillegg at bølgeamplituden ζ_a er kansellert og at man benytter absoluttverdier for H_k .

4.4 Strukturens respons

Overføringsfunksjonene gir oss som nevnt forholdet mellom signal inn ζ_A og signal ut $|H_j|$. Dermed er selve overføringsfunksjonen meningsløs om den ikke blir satt inn i en større sammenheng, nemlig responsligningen (8) som for ordens skyld blir gjentatt her

$$\sum_{i=1}^N A_j |H(\omega_j)| \sin(\omega_j t + \delta(\omega_j) + \epsilon_j)$$

igjen der A_j er en amplitude fra en spesifikk regulær bølge, ω_j er frekvensen for den regulære bølgekomponenten, $\delta(\omega_j)$ er fasevinkelen for legemets respons og ϵ_j er den tilfeldige fasevinkelen for en bølgekomponent.

Fasevinklene $\delta(\omega_j)$ for hiv og stamp kommer fra (19) i utledningen av differensialligningen gjort ovenfor. Det beklages at ϵ_j og $\delta(\omega_j)$ "bytter plass" sammenlignet med de opprinnelige uttrykkene for overflateheving (2) og respons (8). Fasevinkeluttrykkene er

$$\delta_3(\omega_j) = \arctan\left(-\frac{\omega B_{33}}{C_{33} - (M + A_{33})\omega^2}\right) \tag{33}$$

for hiv, og tilsvarende

$$\delta_5(\omega_j) = \arctan\left(\frac{\omega B_{55}}{C_{55} - (I_{55} + A_{55})\omega^2}\right) \tag{34}$$

for stampebevegelsen.

Når det gjelder bølgekomponentens fasevinkel ϵ_j trenger ikke denne beregnes på samme måte, men kan plukkes tilfeldig slik at $\epsilon_j \in [0, 2\pi]$. Matematisk er dette prinsippet gyldig, og fysisk er det heller ikke feil fordi man uansett ikke kan forutsi en sjøgang på forhånd.

Prosjektet burde nå være godt underbygd teoretisk. Resultatene for lekterne er presentert i del 6 av rapporten.

5 Om beregningsprogrammet

5.1 Målsetting med program

Programmet er som tidligere nevnt todelt, en del gjøres i Fortran, og en del gjøres i C. Fortran-delen skal beregne bølgestatistikk, mens C-programmet skal beregne respons og bølgebevegelser og skrive resultatene til en fil som GLView kan lese og kjøre.

Diverse merknader til FORTRAN og C

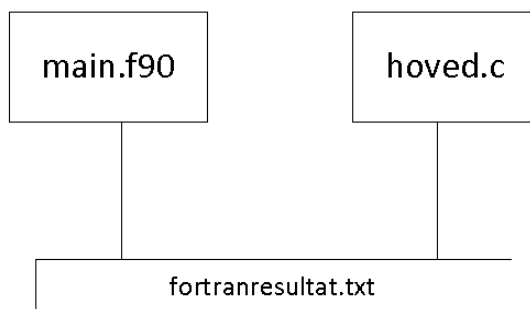
Det ser ut til at forskjellen mellom subrutiner og funksjoner er at man bruker funksjoner til å beregne enkeltverdier gjennom funksjonens navn, mens en subrutine kan dedikere flere verdier til flere variable som er oppgitt i rutinen. I starten ble det brukt subrutiner til å returnere enkeltverdier, men i etterkant ville det vært bedre å bruke funksjoner til disse beregningene. (Hargrove og Whitlock 1996)

5.2 Programstruktur

I dette avsnittet vil det gis en kortfattet forklaring av hvordan programmet er bygd opp. På grunn av omfanget av variable og matematiske operasjoner, er dataflytdiagrammer valgt bort.

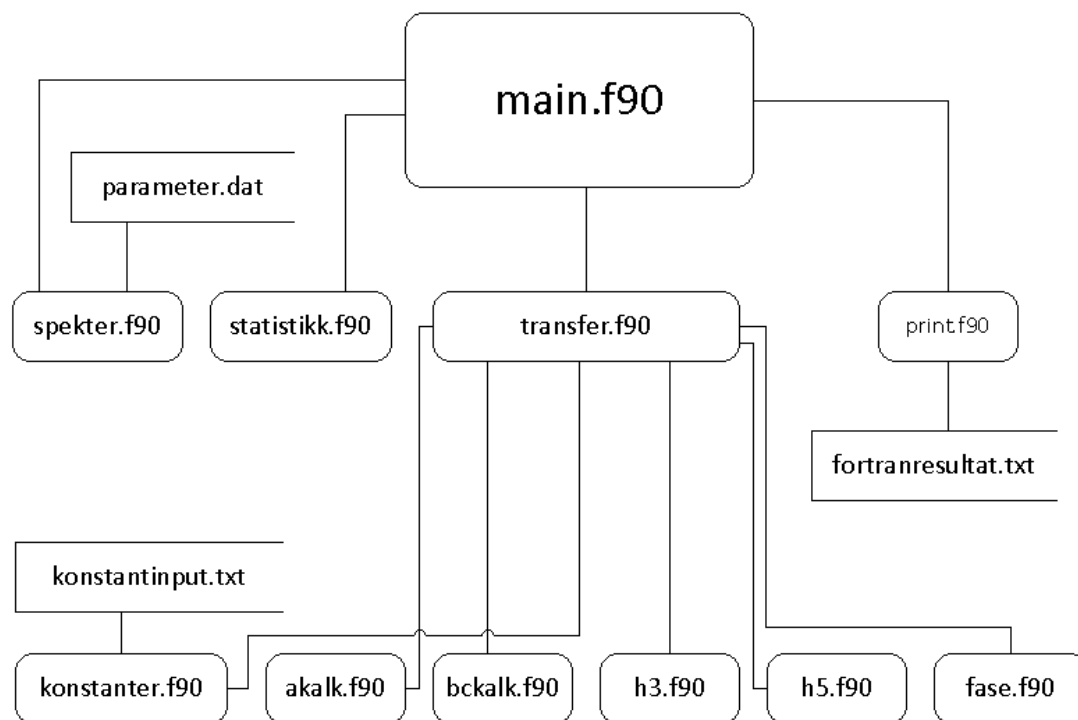
Generelt er det lagt vekt på ingen globale variabler, men stor vekt på resultatvektorer- og matriser som sendes mellom subrutinene. I denne strukturpresentasjonen er en del mindre viktige skrivefiler utelatt.

All kode finnes i egen appendiks.



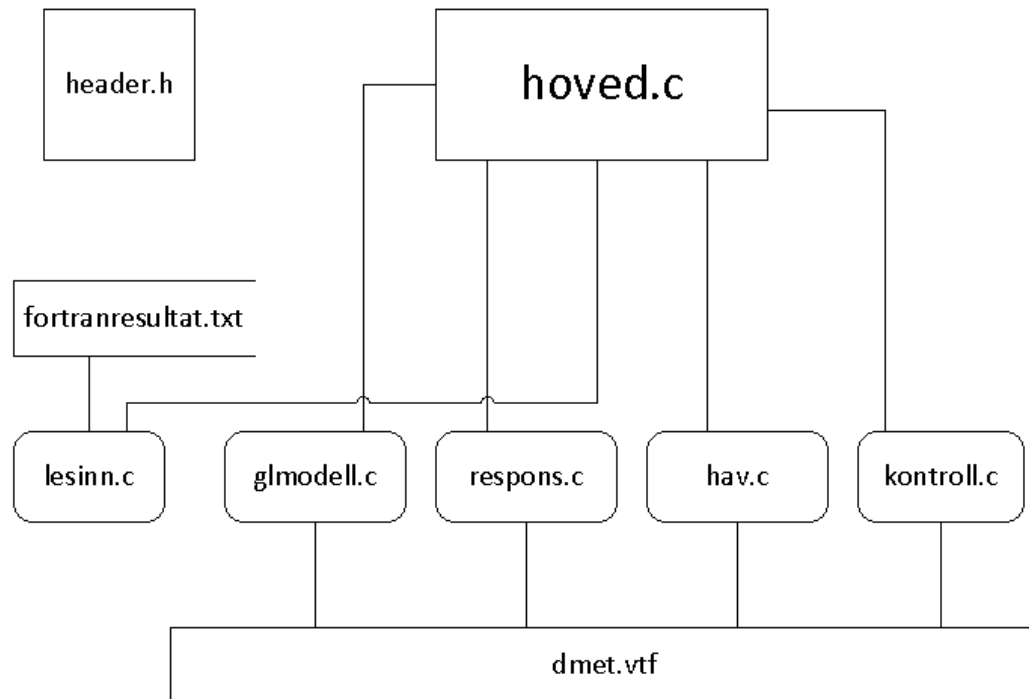
Figur 5: Modell av datastruktur mellom Fortran og C-program

Figur 5 viser hvordan Fortran-programmet og C-programmet samarbeider. Det er ingen direkte kryssfunksjonalitet mellom programmene. Fortran-programmet gjør seg helt ferdig, og skriver alle relevante resultater til en tekstfil. C-programmet leser resultatfilen i tillegg til lektergeometrien, og beregner sin del.



Figur 6: Modell av datastruktur i Fortran-program

Figur 6 viser hvordan Fortran-programmet er bygd opp. Man kan betrakte programmet som bygd opp av tre lag. Hovedprogrammet `main.f90` er ansvarlig for å kalle opp alle de fire subrutinene som gjør hver sine beregningsoperasjoner. Som det fremkommer av samme figur, er det transfer-rutinen som gjør den største jobben. Denne rutinen kaller opp seks egne subrutiner for å kunne beregne lekterbevegelsen. Til sist kaller `main.f90` opp `print`-rutinen, som overleverer resultatene indirekte til C.



Figur 7: Modell av datastruktur i C-program

Figur 7 viser hvordan C-programmet er bygd opp. En header-fil inneholder alle standardbibliotekene som C trenger, i tillegg til å definere en del konstanter som f.eks havteppegeometri og tidssteg for visualisering. Hovedprogramfilen `hoved.c` holder styr på fem subrutiner. Den første subrutinen som kjøres leser inn den nå meget omtalte `fortranresultat.txt`. `Glmodell.c` skriver all grunnleggende geometri for havflate og lektere. `Respons.c` og `hav.c` benytter matematikken til å beregne utslag i tidligere definerte noder, mens `kontroll.c` sammenstiler de tre resultatgruppene til å gjelde for riktig tidssteg.

6 Resultater fra programmet

I denne inndeling vil resultatene som er beregnet og produsert i programmene, presenteres visuelt i form av plott og tabeller. Resultatene diskuteres ikke i denne inndeling, kun presentasjon.

Det er viktig å bemerke seg at lekter A har bredde 10 meter og dypgang 5 meter. Lekter B har bredde 14 meter og dypgang 3.57 meter.

6.1 Bølgestatistikk

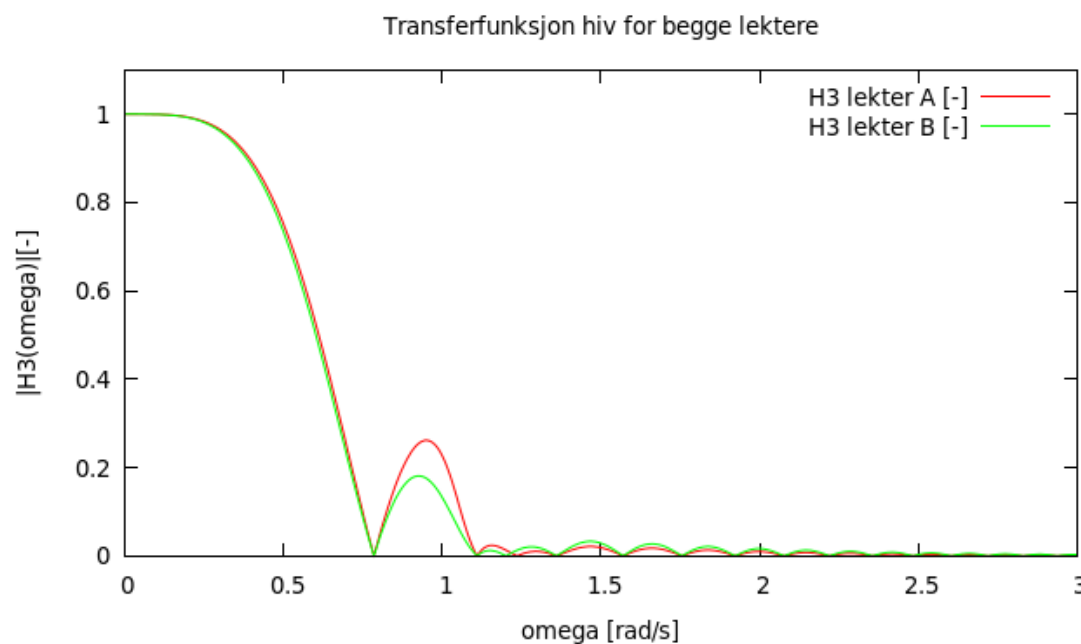
1	-----
2	Nullkrysningsperioden T_{m0z} er: 7.9[s]
3	Midlere periode T_{m0e} er: 8.3[s]
4	Midlere periode mellom bølgetoppene er: 5.7[s]
5	Høyeste bølge i en 5-timers periode er: 23.6[m]
6	-----

Figur 8: Bølgestatistikk som beregnet fra program

Som oppgaven krever, skal ulike verdier for bølgestatistikken oppgis. I figur 8 vises de fire verdiene som skrevet til filen statistikk.txt. Hvordan verdiene er beregnet er tidligere nevnt i teoridelen.

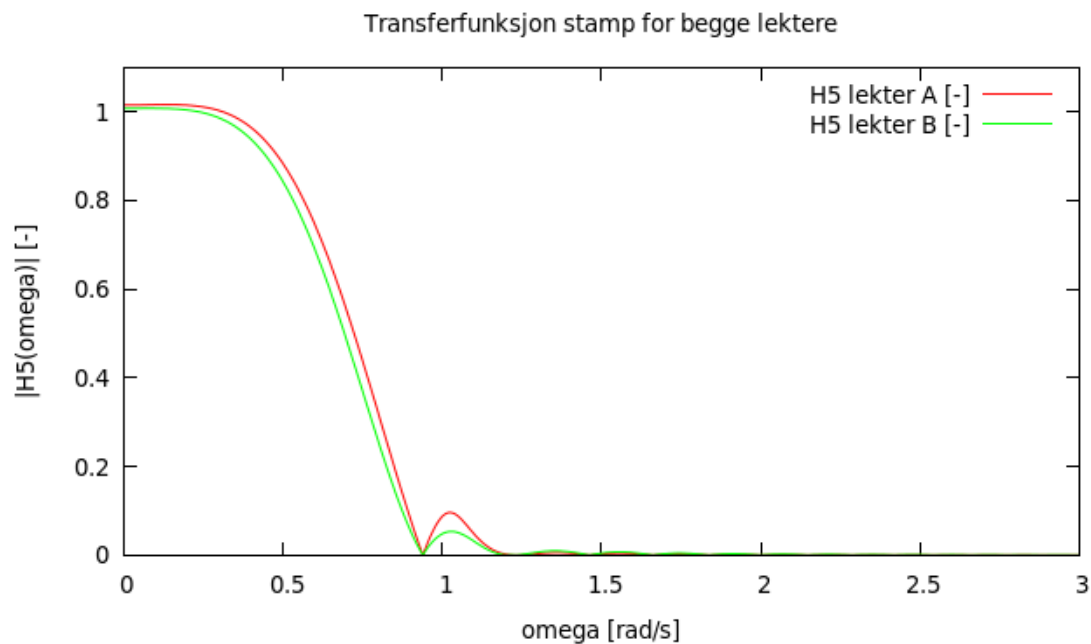
6.2 Beregnede transferfunksjoner for hiv og stamp

Som beskrevet i teoridelen er overføringsfunksjoner forholdet mellom påsatt last og respons. Figur 9 og 10 viser disse som beregnet i Fortran-programmet.



Figur 9: Transferfunksjoner for hiv, begge lektere

Figur 9 viser hvordan leker A og B svarer på bølgekreftene med en hivbevegelse midtskips.

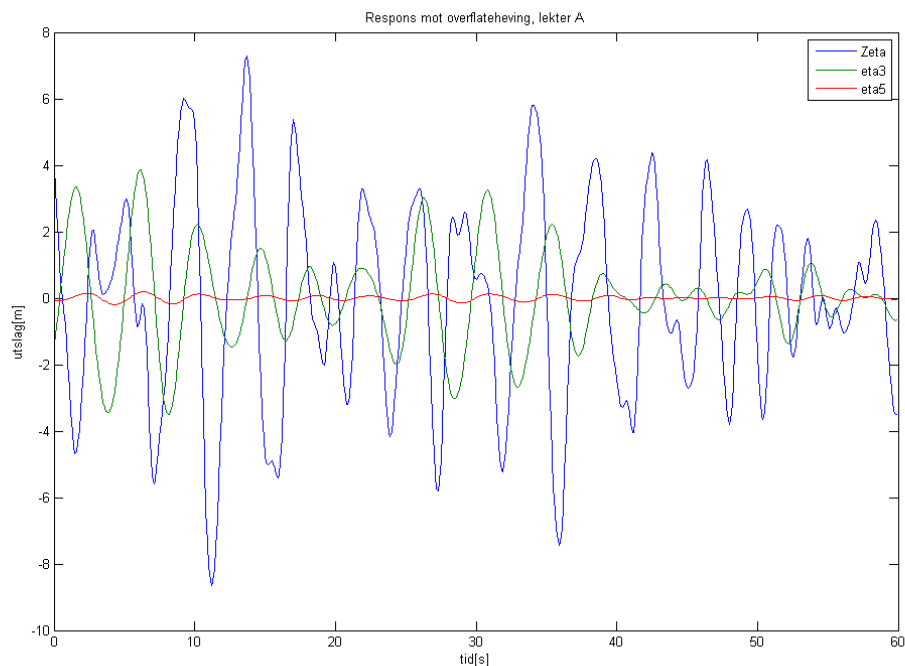


Figur 10: Transferfunksjoner for stamp, begge lektere

Figur 10 viser hvordan leker A og B svarer på bølgekreftene med en stampebevegelse midtskips.

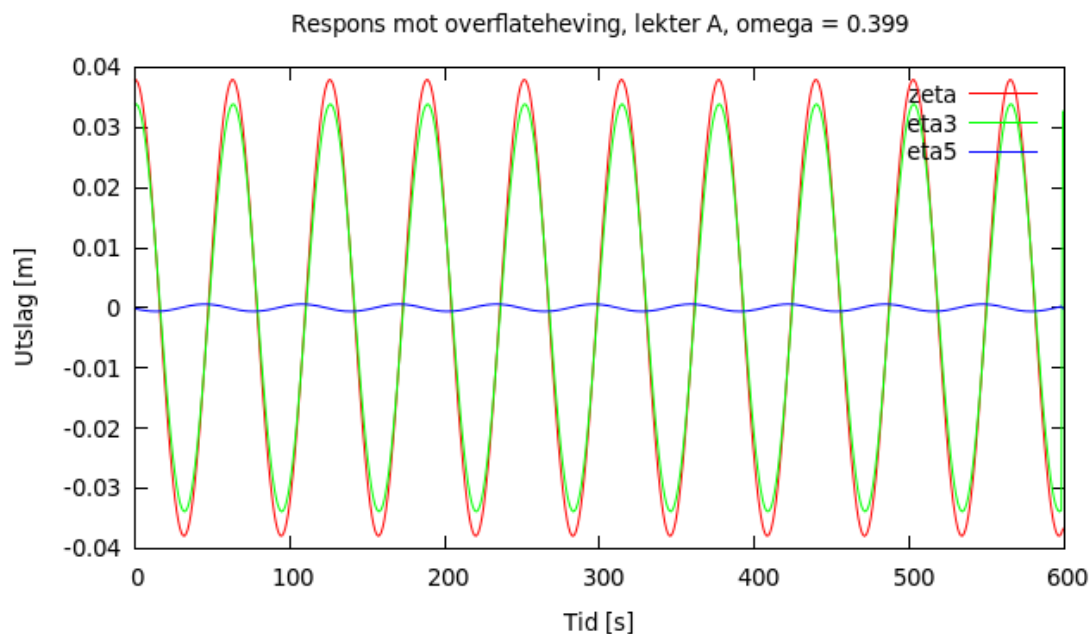
6.3 Respons, havflate og visualisering

I denne inndeling presenteres ikke resultatene fra C-delen direkte, siden disse best ses ved å kjøre dmet.vtf enten vedlagt eller fra programmet. Likevel vises noen illustrative eksempler på at programmet har fungert.



Figur 11: Plott av overflate mot respons irregulær sjø

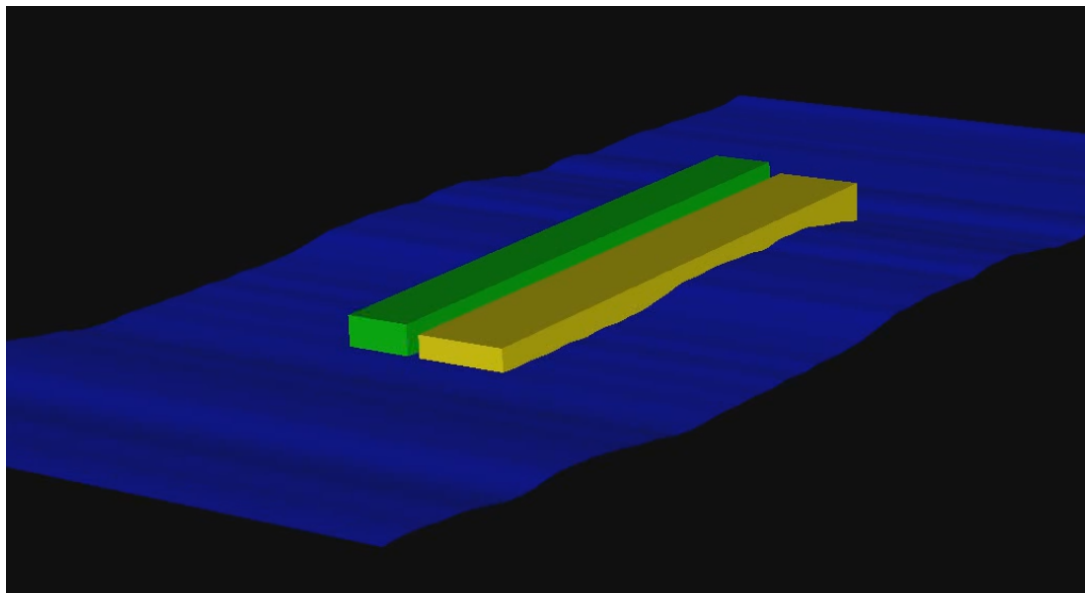
Figur 11 viser et kontrollplott fra en tidlig fase av innkjøringen. I sin irregulære natur, vil det være vanskelig å tolke noe konkret ut fra plottet. Blå linje er overflateheving, mens grønn linje er utslag midtskips hiv.



Figur 12: Respons mot bølge, regulær sjø

Figur 12 viser kanskje en bedre pekepinn på om C-programmet har jobbet riktig. For en frekvens på 0.399 er det plottet overflateheving midtskips zeta, hivbevegelse midtskips eta3, og

stampebevegelse midtskips eta5. Denne figuren kan også tas ut av programmet direkte ved å kjøre vedlagte shell script.

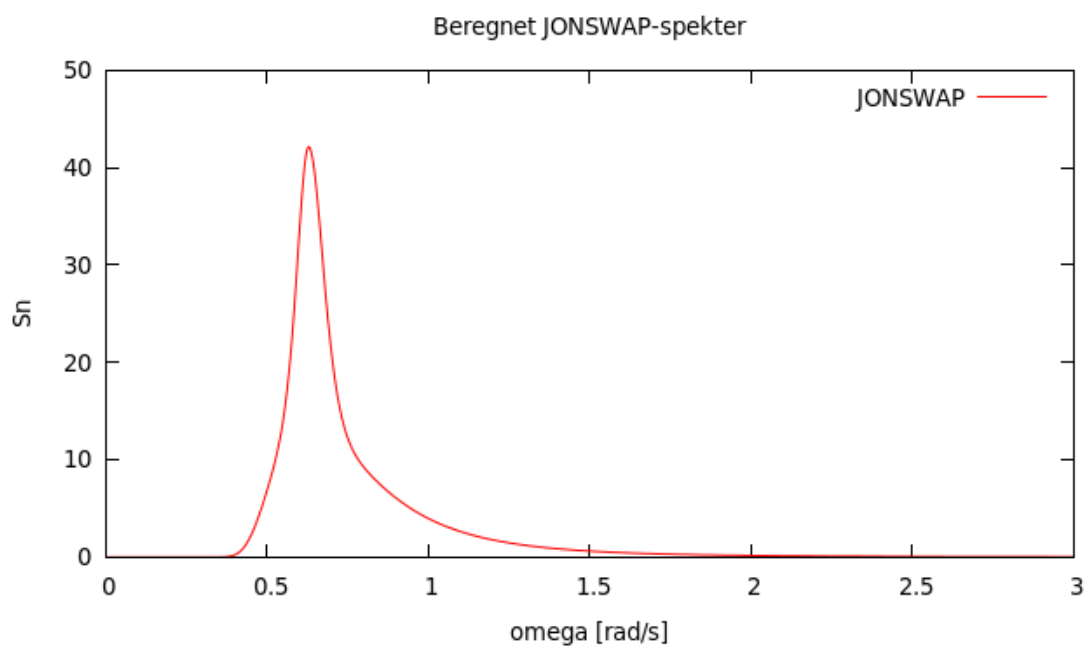


Figur 13: Skjerm bilde av visualisering i GLView

Figure 13 viser et skjermbilde av visualiseringen i GLView. Animasjonen kan reproduseres ved å kjøre VTF-filene som nevnt. Antall bølgekomponenter plukket jevnt over spekteret er 100.

6.4 Andre resultater fra programmet

JONSWAP-spekteret

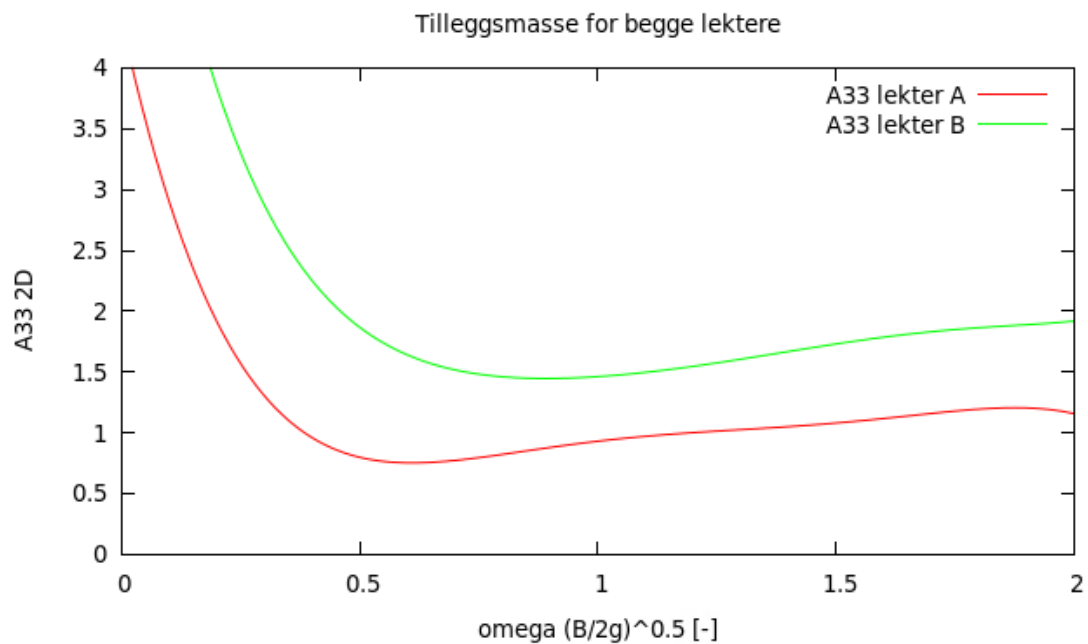


Figur 14: JONSWAP-spekter fra øving 3

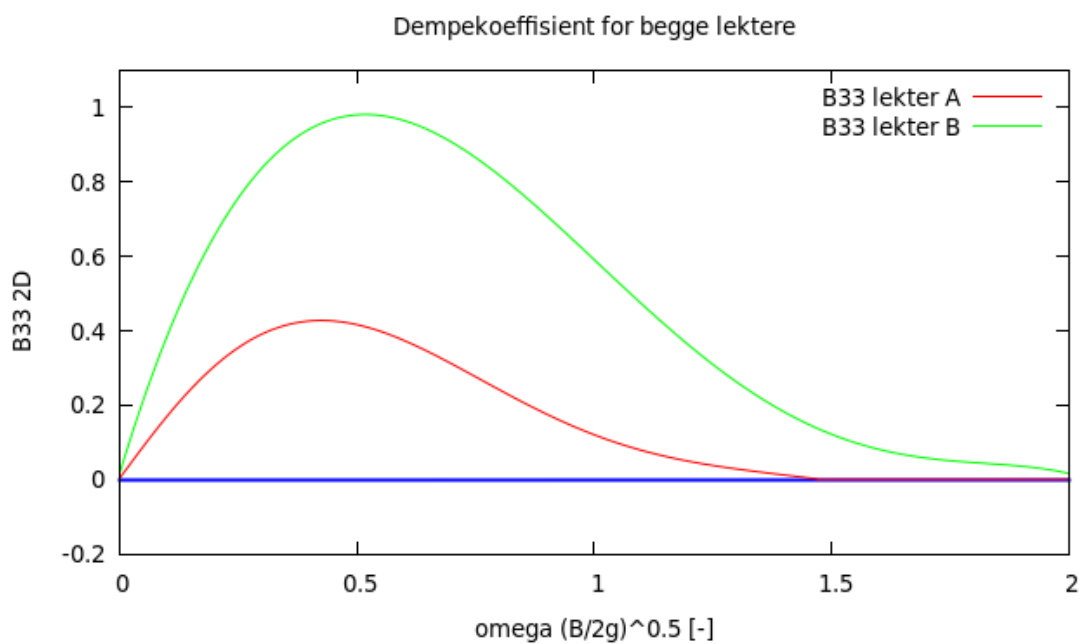
Figur 14 viser bølgespekteret slik det er beregnet i programmet.

Hydrodynamiske koeffisienter

Grafene i (Pettersen 2007, figur 3.20) ble avlest for hånd, og datasettene ble lastet inn i Matlab. Subrutinene akalk.f90 og bckalk.f90 beregnet deretter todimensjonale kurver, og resultatet ble som i figur 15



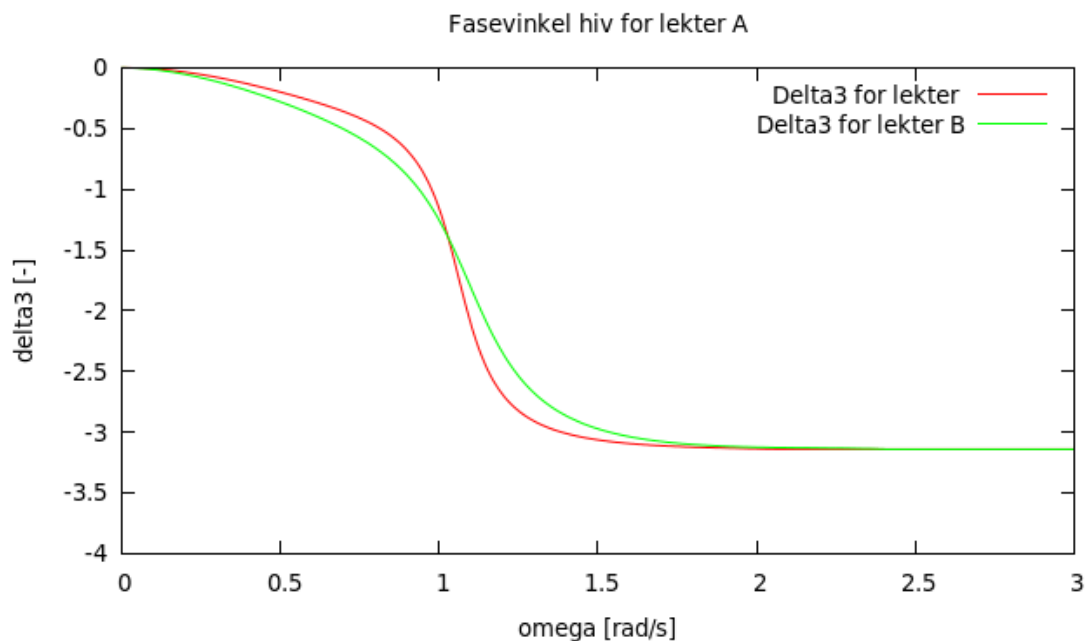
Figur 15: Todimensjonal tilleggsmasse for hiv



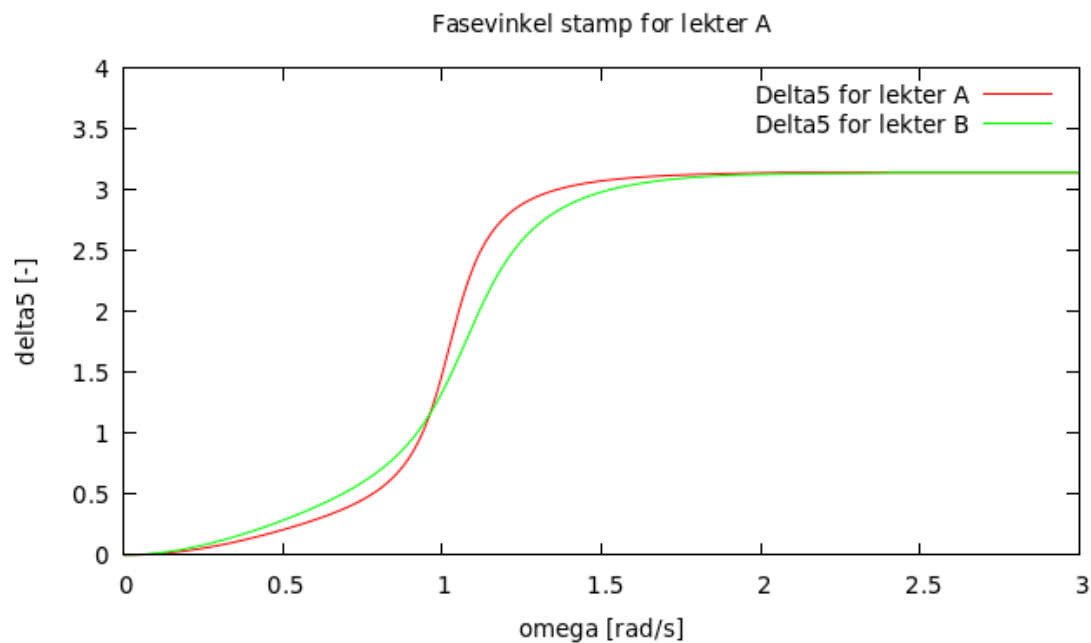
Figur 16: Todimensjonale dempekoeffisienter for hiv

Det er viktig å bemerke at i plottene i figurene i marinkompendiet, så er enhetene dimensjonsløse. For å få ut riktig trdimensjonal verdi må det ganges inn, også med lengden til lekteren siden avlest verdig gjelder kun for en todimensjonal seksjon.

Fasevinkler for hiv og stamp



Figur 17: Fasevinkler for hiv, begge lektere



Figur 18: Fasevinkler for stamp, begge lektere

Figur 17 og 18 viser fasevinklene δ_3 for hiv og δ_5 for stamp slik de er beregnet fra programmet.

7 Diskusjon

7.1 Konsekvens av forenklinger og antagelser

I denne oppgaven er det gjort en rekke forenklinger i forhold til hvordan et sjøbilde hadde sett ut i det virkelige liv. Man har eliminert muligheten for at sjøen kommer fra ulike retninger, i tillegg til å ha kun to frihetsgrader for lekterne. I tillegg ser man bort fra rulling, noe som ville vært en naturlig respons på sjø fra flere retninger.

Sett i forhold til kompleksiteten i oppgaven virker det likevel som en svært fornuftig forholdsregel å gjøre de nevnte antagelser.

7.2 Bølgespekter og feilkilder

Det kan se ut som Fortran-programmet har regnet riktig verdi for spekteret i følge 14. Dermed er det grunn til å anta at også amplitudeverdiene for bølgene burde være riktig. Sjøtilstanden vi har fått oppgitt er relativt grov, i og med en signifikant bølgehøyde på 12 meter, og over 23 meters makshøyde (sannsynligvis) over en femtimers periode.

7.3 Lekterrespons og feilkilder

Ut fra plottene for transferfunksjoner 9 og 10 virker det som om Fortran-programmet har gjort en relativt bra jobb. For frekvensverdier mot 0 ser man at verdien går mot 1, fordi at for veldig lange (og små) bølger vil lekteren naturlig nok bevege seg som bølgene. Egenfrekvensen ser ut til å være i underkant av 1, noe som kan stemme ut fra beregningene.

I tillegg virker det som om C-programmet også gjør en fornuftig jobb, i alle fall ut fra 12, men det er fortsatt en usikkerhet om fasevinklene har riktig fortegn. Animasjonen viser kanskje at lekteren er litt lett i hekken, men det er usikkert om dette kun skyldes den svært grove sjøtilstanden vi har fått oppgitt.

7.4 Generalisering av kode

Generalisering av koden fungerer i utgangspunktet for flere lektere, men det som ikke er generalisert er polynomene for de hydrodynamiske koeffisientene. Disse er hardkodete for de to lekterne fordi utgangspunktet er kurvene i marinkompndiet. Man kunne heller brukt en tilnærming, alternativt brukt en del tid på å lage en interpolasjonsfunksjon som beregnet koeffisientene for flere lektere med ulik bredde/ dypgangsforhold.

8 Konklusjon og videre arbeid

Ut fra resultatene og animasjonen, virker det som at ingen av lekterne er sjødyktige for slike sjøtilstander som oppgitt. Det frarådes derfor å igangsette bygging av disse med geometrien som er valgt for denne oppgave.

Videre arbeid bør innbefatte en solid økning av fribord, i tillegg til at man burde vurdert å montere rekker eller overbygg for sikring av last og eventuelt personell om bord. I tillegg bør det spesifiseres om lekterne skal ligge i Nordsjøen hele året, eller kun i sommersesong.

Bokkilder

- Faltinsen, Odd M. (1990). *Sea loads on ships and offshore structures*. Cambridge University Press.
- Pettersen, Bjørnar (2007). *Marin Teknikk 3 kompendie*. Akademika Forlag.
- Myrhaug, D. og W. Lian (2009). *Marine Dynamics lecture notes*. Akademika Forlag.

Andre kilder

- Hargrove, Paul H. og Sarah T. Whitlock (1996). *Fortran 90 tutorial*. URL: https://web.stanford.edu/class/me200c/tutorial_90/08_subprograms.html.
- Bøckmann, Eirik (2008). *Prosjektoppgaven i faget TMR4160 Datametoder for marine applikasjoner*. URL: <http://folk.ntnu.no/eirikbo/>.
- Morin, David (2009). *Oscillations*. URL: <http://www.people.fas.harvard.edu/~djmorin/waves/oscillations.pdf>.
- Holm, Håvard (2015). *TMR4160 Forelesning*.

A Programkode

NB! Programmet kjører kun i Linux.

Ved eventuelle problemer, ring support 4666 9001.

A.1 Bruksanvisning for DMET DELUXE 2000

Som klippet fra LESMEG.TXT:

1. Åpne terminal/ shell/ bash i Linux
2. Naviger til riktig mappe (programfiler")
3. Skriv — ./dmet.sh — (uten strek) i terminal
4. Følg instruksjer på skjermen

eventuelt for kun å kompilere kode i shell/ bash:

1. Naviger til riktig mappe (programfiler")
2. Skriv — make — for å kompilere Fortran-program
3. Skriv — ./fprogram — for å kjøre Fortran-program
4. Skriv — make c — for å kompilere C-program
5. Skriv — ./cprogram — for å kjøre C-program

Visualisering

Det er lagt ved to versjoner av VTF-filen. Den ene filen har Windows-linjeendinger, og er den som skal kjøres på Windows. Filene er klare til å kjøres.

A.2 Vedlagt kode

Listings

1	Fortran-program main.f90	26
2	Subrutine spekter.f90	27
3	Subrutine statistikk.f90	29
4	Subrutine transfer.f90	32
5	Subrutine konstanter.f90	34
6	Subrutine akalk.f90	36
7	Subrutine bckalk.f90	39
8	Subrutine h3kalk.f90	41
9	Subrutine h5kalk.f90	44
10	Subrutine fase.f90	46
11	Program hoved.c	48
12	Subrutine lesinn.c	50
13	Subrutine glmodell.c	52
14	Subrutine respons.c	54

15	Subrutine hav.c	57
16	Subrutine kontroll.c	60
17	header.h	61
18	dmet.sh	63

Listing 1: Fortran-program main.f90

```

1  ! -----
2  !  DMET DELUX 2000 Program      main                      No.:
3  ! -----
4  !  Hensikt :
5  !  Beregne bølgespekter, bølgestatistikk og transfer-funksjoner
6  !
7  !  Metode :
8  !  Definerer parametere som programmet trenger, og kaller på
   subrutiner suksessivt.
9  !
10 !  Kallesekvens:
11 !  "dmet.sh" i terminal/ shell/ kommandolinje
12 !
13 !  Parametre:
14 !  Navn   I/O Type           Innhold/Beskrivelse
15 !
   .....
16 !  omega I/O double           bølgefrequens
17 !  a     I/O flyttall          nederste frekvensverdi
18 !  b     I/O flyttall          øverste frekvensverdi
19 !  n     I/O heltall           antall frekvensintervaller
20 !  n1    I/O heltall           antall lektere
21 !  Hs    I/O heltall           signifikant bølgehøyde
22 !  k     I/O flyttall          bølgetall
23 !  zeta_an I/O flyttalls matrise bølgeamplitude
24 !  sn     I/O flyttalls matrise spekterverdi
25 !  H3     I/O flyttalls matrise transferfunksjon hiv
26 !  H5     I/O flyttalls matrise transferfunksjon stamp
27 !  delta3  I/O flyttalls matrise fasevinkel hiv
28 !  delta5  I/O flyttalls matrise fasevinkel stamp
29 !  A33     I/O flyttalls matrise tilleggsmasse hiv
30 !  A55     I/O flyttalls matrise tilleggsmasse stamp
31 !  B33     I/O flyttalls matrise demping hiv
32 !  B55     I/O flyttalls matrise demping stamp
33 !  L       I/O heltalls vektor  lengde på lekter
34 !  B       I/O heltalls vektor  bredde på lekter
35 !  D       I/O flyttalls vektor  dypgang på lekter
36 !  H       I/O flyttalls vektor  dybde i riss for lekter
37 !  M       I/O flyttalls vektor  masse lekter
38 !  GML     I/O flyttalls vektor  langskips metasenter
39 !  I33     I/O flyttall vektor   andre arealmoment tverrskips
40 !  I55     I/O flyttalls vektor  andre arealmoment langskips
41 !  pi      I/O flyttall          pi
42 !  rho     I/O flyttall          tetthet sjøvann
43 !  g       I/O flyttall          tyngdeakselerasjonen
44 !
45 !  INTERNE VARIABLE:
46 !  Ingen interne variable i hovedprogrammet
47 !

```

```

48 !      SUBROUTINER:
49 !      spekter      Beregner et JONSWAP-spekter
50 !      statistikk   Beregner bølgestatistikk som påkrevd
51 !      transfer     Beregner transferfunksjoner i hiv og stamp
52 !      utskrift     Samle resultater og skrive til fil
53 !
54 !      Programmert av:      Tor Erik Larsen
55 !      Dato/Versjon   :      27.04.15 / 1.0
56 !
57 ! -----
58
59 PROGRAM main
60 IMPLICIT NONE
61
62 !Erklærer variable og parametere brukt i programmet
63 INTEGER,PARAMETER::n=1000, nl=2
64 REAL::a,b,Hs
65 REAL,PARAMETER::pi=3.1415,g=9.81,rho=1025
66 REAL,DIMENSION(1:nl)::L,Br,D,H,M,GML,I33,I55
67 REAL,DIMENSION(1:n,1:nl)::A33,A55,B33,B55
68 DOUBLE PRECISION,DIMENSION(1:n)::omega,sn,zeta_an,k
69 DOUBLE PRECISION,DIMENSION(1:n,1:nl)::H3,H5,delta3,delta5
70
71 !Kaller opp alle subrutiner
72 CALL spekter(a,b,n,pi,omega,sn,zeta_an,k,g,Hs)
73 CALL statistikk(a,b,n,omega,Sn,Hs,pi)
74 CALL transfer(a,b,omega,sn,pi,rho,g,n,nl,L,Br,D,H,M,GML,I33,I55, &
75             A33,A55,B33,B55,k,H3,H5,delta3,delta5)
76 CALL utskrift(n,nl,omega,zeta_an,k,H3,H5,delta3,delta5)
77
78 END PROGRAM

```

Listing 2: Subrutine spekter.f90

```

1 !      -----
2 !      DMET DELUX 2000 Subrutine spekter                      No.:
3 !      -----
4 !      HENSIKT :
5 !      Beregne JONSWAP-spekter, bølgeamplituder og bølgetall
6 !
7 !      METODE :
8 !      Tar inn frekvens og spekterparametere som signifikant bølgehøyde,
9 !      midlere periode og toppethetsfaktor (gamma). Benytter disse for
10 !      å beregne spekterverdier som kan plottes, samt bølgeamplituder
11 !      og bølgetall for hver enkelt frekvens
12 !
13 !      KALLESEKVENNS:
14 !      CALL spekter(a,b,n,pi,omega,sn,zeta_an,k,g,Hs)
15 !
16 !      PARAMETERE:
17 !      Navn   I/O Type      Innhold/Beskrivelse
18 !
19 !      .....
20 !      omega I flyttalls vektor   bølgefrequens
21 !      n     I heltall           antall frekvensintervaller

```



```

22 !   zeta_an I flyttalls vektor   bølgeamplitude
23 !   hs      I heltall           signifikant bølgehøyde
24 !   tp      I heltall           midlere periode
25 !   gamma I heltall           toppethetsfaktor
26 !   pi      I flyttall          pi
27 !   g       I flyttall          tyngdelfeltsakselerasjon
28 !   sn      0 flyttall          spekterverdi
29 !   a       I heltall           nedre frekvensverdi
30 !   b       I heltall           øvre frekvensverdi
31 !
32 !   INTERNE VARIABLE:
33 !   Navn      Funksjon
34 !
35 !   .....
36 !   i,m,n      tellevariabler
37 !   omegap     mellomregningsvariabel
38 !   konstant,v1,v2,v3 mellomregningsvariabler
39 !   deltaomega mellomregningsvariabel
40 !   sigma      frekvensparameter
41 !   omegap      frekvensparameter
42 !
43 !   SUBROUTINER:
44 !   Navn      Funksjonsbeskrivelse
45 !
46 !   .....
47 !   LESEFILER:
48 !   Navn      Eventuell beskrivelse
49 !
50 !   .....
51 !   parameter.dat inneholder spekterinformasjon oppgitt
52 !
53 !   SKRIVEFILER:
54 !   Navn      Eventuell beskrivelse
55 !
56 !   .....
57 !   spekterplot.dat skriver frekvens og spekterverdier
58 !
59 !   -----
60 !
61 !   Programmert av:      Tor Erik Larsen
62 !   Dato/Versjon   :      27.04.15 / 1.0
63 !
64 !   -----
65
66 SUBROUTINE spekter(a,b,n,pi,omega,sn,zeta_an,k,g,Hs)
67 IMPLICIT NONE
68
69 !Erklærer variable
70 INTEGER::i,m,n,hs
71 REAL::a,b,tp,gamma,omegap,pi,g,deltaomega
72 DOUBLE PRECISION::konstant,v1,v2,v3
73 DOUBLE PRECISION,DIMENSION(1:n)::omega,sn,sigma,zeta_an,k
74

```

```

75 !opner inndatafil og regner omegap, skriver kontroll til skjerm
76 OPEN(10,file="parameter.dat")
77 READ(10,*) hs,tp,gamma,a,b
78 CLOSE(10)
79
80 !Beregner omega-vektor
81 m = SIZE(omega,1)
82 DO i=1,m
83     omega(i) = a + (b-a)*(i-0.5)/n
84 END DO
85
86 !Beregner omegap
87 omegap = (2.0*pi)/tp
88
89 !Beregner sigma-vektor
90 do i = 1,n
91     if (omegap < omega(i)) then
92         sigma(i) = 0.09
93     ELSE
94         sigma(i) = 0.07
95     END IF
96 END DO
97
98 !Faktorerer Sn
99 DO i = 1,n
100     konstant = (5.0/(32.0*pi))*(hs**2)*tp*(1.0 - 0.287*log(gamma))
101     v1 = (omegap/omega(i))**5
102     v2 = exp(-(5.0/4.0)*((omegap/omega(i))**4))
103     v3 = gamma**(exp(-((omega(i)/omegap)-1.0)**2/(2.0*sigma(i)**2)))
104     sn(i)=konstant*v1*v2*v3
105 END DO
106
107 !Beregner amplitudevektor
108 deltaomega = (b-a)/n
109 DO i=1,n
110     zeta_an(i) = SQRT(2.0*sn(i)*deltaomega)
111 END DO
112
113 !Regner ut vektor for bølgetallet k
114 DO i=1,n
115     k(i)=omega(i)**2 / g
116 END DO
117
118 !Skriver spekteret til fil som kan plottes
119 OPEN(1010,FILE="spekterplot.dat",ACTION="WRITE",STATUS="UNKNOWN")
120 DO i=1,n
121     WRITE(1010,*) omega(i), sn(i)
122 END DO
123 CLOSE(1010)
124
125 !Gir tilbake kontrollen og avslutter subrutine
126 RETURN
127 END SUBROUTINE

```

Listing 3: Subrutine statistikk.f90

```

1  ! -----
2  !  DMET DELUX 2000 Subrutine statistikk
3  ! -----
4  !  HENSIKT :
5  !  Beregner bølgestatistikk fra oppgitt JONSWAP-spekter
6  !
7  !  METODE :
8  !  Tar inn frekvensvektor, spekterverdier og spekterparametere fra
9  !  tidligere subrutiner. Beregner spektermomenter, midlere periode
10 !  og maksimal bølgehøyde for en femtimers periode.
11 !
12 !  KALLESEKVENSS:
13 !  CALL statistikk(a,b,n,omega,Sn,Hs,pi)
14 !
15 !  PARAMETERE:
16 !  Navn   I/O Type           Innhold/Beskrivelse
17 !  .....
18 !  omega I flyttalls vektor   bølgefrequens
19 !  n     I heltall            antall frekvensintervaller
20 !  k     I flyttallsvektor    bølgetall
21 !  zeta_an I flyttallsvektor  bølgeamplitude
22 !  hs     I heltall            signifikant bølgehøyde
23 !  tp     I heltall            midlere periode
24 !  gamma I heltall            toppethetsfaktor
25 !  pi     I flyttall          pi
26 !  g      I flyttall          tyngdelfeltsakselerasjon
27 !  sn     I flyttallsvektor    spekterverdi
28 !  a      I heltall            nedre frekvensverdi
29 !  b      I heltall            øvre frekvensverdi
30 !  pi     I flyttall          pi
31 !  rho    I flyttall          tetthet sjøvann
32 !  g      I flyttall          tyngdeakselerasjonen
33 !  Tm02   0 flyttall          nullkrysningsperiode
34 !  Tm0e   0 flyttall          middelperiode
35 !  Tm24   0 flyttall          middelperiode mellom topper
36 !  Hmax   0 flyttall          største bølgehøyde fem timer
37 !
38 !  INTERNE VARIABLE:
39 !  Navn           Funksjon
40 !
41 !  .....
42 !  i,n            tellevariabler
43 !  deltaomega     frekvensintervall
44 !  m0,m1,m2,m4    spektermomenter
45 !  mndelsum       mellomregningsverdier
46 !
47 !  SUBROUTINER:
48 !  Navn           Funksjonsbeskrivelse
49 !  .....
50 !  konstanter     beregner metasenter og andre arealmomenter
51 !  akalk           beregner tilleggmasse i hiv
52 !  bckalk          beregner demping og fjærkoeffisienter
53 !  h3kalk          beregner transferfunksjoner hiv
54 !  h5kalk          beregner transferfunksjoner stamp
55 !  fasekalk        beregner fasevinkel for bevegelser

```

```

55 !
56 !   LESEFILER:
57 !   Navn           Eventuell beskrivelse
58 !
59 !   .....
59 !   konstantinput.txt   inneholder lektergeometri
60 !
61 !
62 !   SKRIVEFILER:
63 !   Navn           Eventuell beskrivelse
64 !
65 !   .....
65 !   statistikk.txt       resultater for bølgestatistikk
66 !
67 !   -----
68 !
69 !   Programmert av:      Tor Erik Larsen
70 !   Dato/Versjon   :      27.04.15 / 1.0
71 !
72 !   -----
73
74 SUBROUTINE statistikk(a,b,n,omega,Sn,Hs,pi)
75 IMPLICIT NONE
76 INTEGER i,n
77 REAL::m0delsum, m1delsum,m2delsum,m4delsum
78 REAL::m0,m1,m2,m4
79 REAL::Tz,Tm02,Tmo2,Tm24,Hmax,NN,Tmoe
80 REAL::pi,Hs,a,b,deltaomega
81 DOUBLE PRECISION,DIMENSION(1:n)::omega,sn
82
83 !Beregner deltaomega
84 deltaomega = (b-a)/n
85
86 !Beregner spektermomentet m0
87 m0delsum = 0
88 DO i = 1,n
89     m0delsum = m0delsum + sn(i)
90 END DO
91 m0 = deltaomega * m0delsum
92
93 !Beregner spektermomentet m1
94 m1delsum = 0
95 DO i = 1,n
96     m1delsum = m1delsum + sn(i)*omega(i)
97 END DO
98 m1 = deltaomega*m1delsum
99
100 !Beregner m2 og tm02
101 m2delsum = 0
102 DO i=1,n
103     m2delsum = m2delsum + omega(i)**2 * sn(i)
104 END DO
105 m2 = deltaomega*m2delsum
106
107 !Beregner spektermomentet m4
108 m4delsum = 0
109 DO i=1,n

```

```

110     m4delsum = m4delsum + omega(i)**4 * sn(i)
111 END DO
112 m4 = deltaomega*m4delsum
113
114 !Beregner nullkrysningsperiode Tz
115 Tz = pi*Hs / (2.0*SQRT(m2))
116 Tm02= 2.0*pi*SQRT(m0/m2)
117 Tmoe = 2.0*pi*(m0/m1)
118 Tm24 = 2.0*pi*SQRT(m2/m4)
119
120 !Beregner antall bølger N for en varighet på 5 timer eller 18000
    sekunder
121 NN = 18000.0 / Tz
122 Hmax = Hs*SQRT(LOG(NN)/2.0)
123
124 !Skriver resultater til fil
125 OPEN(252,FILE="statistikk.txt",ACTION="WRITE",STATUS="UNKNOWN")
126 WRITE(*,*) ''
127 WRITE(252,*) '-----'
128 WRITE(252,"(A,F10.1,A)") 'Nullkrysningsperioden Tm0z er: ', Tm02, '[s]'
129 WRITE(252,"(A,F10.1,A)") 'Midlere periode Tm0e er: ', Tmoe, '[s]'
130 WRITE(252,"(A,F10.1,A)") 'Midlere periode mellom bølgetoppene er:
    ',Tm24, '[s]'
131 WRITE(252,"(A,F10.1,A)") 'Høyeste bølge i en 5-timers periode er:
    ',Hmax, '[m]'
132 WRITE(252,*) '-----'
133 WRITE(*,*) ''
134 CLOSE(252)
135
136 RETURN
137 END SUBROUTINE

```

Listing 4: Subrutine transfer.f90

```

1  ! -----
2  !  DMET DELUX 2000 Subrutine transfer                      No.:
3  ! -----
4  !  HENSIKT :
5  !  Samordne alle subrutiner som trengs for å beregne transferfunksjoner
6  !  for lekterne
7  !
8  !  METODE :
9  !  Tar inn parametere og konstanter tidligere definert i hovedprogram.
10 !  Kaller opp subrutiner suksessivt for å beregne transferfunksjoner
11 !  og fasevinkler for begge lektere samtidig. Teorien bak beregningene
12 !  vises i rapporten. Grunnen til at alle parametere er erklært i denne
13 !  subrutinen er at alle resultater sendes som vektorer inn og ut.
14 !
15 !  KALLESEKVENSS:
16 !  CALL transfer(a,b,omega,sn,pi,rho,g,n,nl,L,Br,D,H,M,GML,I33,I55, &
17 !      A33,A55,B33,B55,k,H3,H5,delta3,delta5)
18 !
19 !  PARAMETERE:
20 !  Navn   I/O Type          Innhold/Beskrivelse
21 !
    .....

```

```

22 ! omega I flyttalls vektor   bølgefrequens
23 ! n     I heltall           antall frekvensintervaller
24 ! nl    I/O heltall         antall lektere
25 ! k     I flyttalls vektor   bølgetall
26 ! zeta_an I flyttalls vektor bølgeamplitude
27 ! hs    I heltall           signifikant bølgehøyde
28 ! tp    I heltall           midlere periode
29 ! gamma I heltall           toppethetsfaktor
30 ! pi     I flyttall          pi
31 ! g      I/O flyttall        tyngdelfeltsakselerasjon
32 ! sn     0 flyttall          spekterverdi
33 ! a      I/O heltall         nedre frekvensverdi
34 ! b      I/O heltall         øvre frekvensverdi
35 ! H3     I/O flyttalls matrise transferfunksjon hiv
36 ! H5     I/O flyttalls matrise transferfunksjon stamp
37 ! delta3 I/O flyttalls matrise fasevinkel hiv
38 ! delta5 I/O flyttalls matrise fasevinkel stamp
39 ! A33    I/O flyttalls matrise tilleggsmasse hiv
40 ! A55    I/O flyttalls matrise tilleggsmasse stamp
41 ! B33    I/O flyttalls matrise demping hiv
42 ! B55    I/O flyttalls matrise demping stamp
43 ! C33    I/O flyttalls vektor fjørkoeffisient hiv
44 ! C55    I/O flyttalls vektor fjørkoeffisient stamp
45 ! L      I/O heltalls vektor lengde på lekter
46 ! B      I/O heltalls vektor bredde på lekter
47 ! D      I/O flyttalls vektor dypgang på lekter
48 ! H      I/O flyttalls vektor dybde i riss for lekter
49 ! M      I/O flyttalls vektor masse lekter
50 ! GML    I/O flyttalls vektor langskips metasenter
51 ! I33    I/O flyttall vektor andre arealmoment tverrskips
52 ! I55    I/O flyttalls vektor andre arealmoment langskips
53 ! pi     I/O flyttall          pi
54 ! rho    I/O flyttall          tetthet sjøvann
55 ! g      I/O flyttall          tyngdeakselerasjonen
56 !
57 !
58 ! INTERNE VARIABLE:
59 ! Navn          Funksjon
60 !
61 ! .....
62 ! i,m,n          tellevariabler
63 ! omegap         mellomregningsvariabel
64 ! konstant,v1,v2,v3 mellomregningsvariabler
65 ! deltaomega     mellomregningsvariabel
66 ! sigma          frekvensparameter
67 ! omegap         frekvensparameter
68 !
69 ! SUBROUTINER:
70 ! Navn          Funksjonsbeskrivelse
71 ! .....
72 ! konstanter     beregner metasenter og andre arealmomenter
73 ! akalk          beregner tilleggsmasse i hiv
74 ! bckalk         beregner demping og fjørkoeffisienter
75 ! h3kalk         beregner transferfunksjoner hiv
76 ! h5kalk         beregner transferfunksjoner stamp
77 ! fasekalk       beregner fasevinkel for bevegelser

```

```

77 ! -----
78 !
79 !   Programmert av:      Tor Erik Larsen
80 !   Dato/Versjon   :      27.04.15 / 1.0
81 !
82 ! -----
83
84 SUBROUTINE transfer(a,b,omega,sn,pi,rho,g,n,nl,L,Br,D,H,M,GML,I33,I55, &
85     A33,A55,B33,B55,k,H3,H5,delta3,delta5)
86 IMPLICIT NONE
87
88 !Erklærer variable og parametere
89 INTEGER::n,nl
90 REAL::hs,tp,gamma,a,b,pi,g,rho
91 REAL,DIMENSION(1:n)::L,Br,D,H,M,GML,I33,I55,omega,k
92 REAL,DIMENSION(1:nl)::C33,C55
93 REAL,DIMENSION(1:n,1:nl)::A33,A55,B33,B55
94 DOUBLE PRECISION,DIMENSION(1:n,1:nl)::H3,H5,delta3,delta5
95
96 !Kaller opp subrutinene
97 CALL konstanter(L,Br,D,H,M,GML,I33,I55,rho,g,nl)
98 CALL akalk(n,nl,rho,g,omega,L,Br,D,A33,A55)
99 CALL bckalk(n,nl,rho,g,omega,L,Br,D,GML,B33,B55,C33,C55)
100 CALL h3kalk(k,omega,n,nl,rho,g,A33,B33,C33,L,Br,D,M,H3)
101 CALL h5kalk(n,nl,L,Br,D,M,A33,B33,A55,B55,C55,I55,k,omega,rho,g,H5)
102 CALL fasekalk(n,nl,omega,M,A33,B33,C33,A55,B55,C55,I55,delta3,delta5)
103
104 RETURN
105 END SUBROUTINE

```

Listing 5: Subrutine konstanter.f90

```

1 ! -----
2 !   DMET DELUX 2000 Subrutine konstanter
3 ! -----
4 !   HENSIKT :
5 !   Beregner konstantverdier for begge lektene, slik som andre
6 !   arealmoment og metasenter
7 !
8 !   METODE :
9 !   Leser lektergeometri fra inputfil, og beregner andre arealmoment,
10 !   og begge metasentre for begge lektene. Sender verdiene tilbake til
11 !   transfer-subrutine
12 !
13 !   KALLESEKVENNS:
14 !   CALL konstanter(L,Br,D,H,M,GML,I33,I55,rho,g,nl)
15 !
16 !   PARAMETERE:
17 !   Navn   I/O Type           Innhold/Beskrivelse
18 !   .....
19 !   n      I heltall           antall frekvensintervaller
20 !   nl     I/O heltall         antall lektene
21 !   g      I/O flyttall        tyngdelfeltsakselerasjon
22 !   L      I/O heltalls vektor  lengde på lekter
23 !   B      I/O heltalls vektor  bredde på lekter
24 !   D      I/O flyttalls vektor dypgang på lekter

```

```

25 ! H I/O flyttalls vektor dybde i riss for lekter
26 ! M I/O flyttalls vektor masse lekter
27 ! GML I/O flyttalls vektor langskips metasenter
28 ! I33 I/O flyttall vektor andre arealmoment tverrskips
29 ! I55 I/O flyttalls vektor andre arealmoment langskips
30 ! pi I/O flyttall pi
31 ! rho I/O flyttall tetthet sjøvann
32 ! g I/O flyttall tyngdeakselerasjonen!
33 !
34 ! INTERNE VARIABLE:
35 ! Navn Funksjon
36 !
37 ! .....
38 ! i,j tellevariabler
39 ! KG tyngdepunkt lekter
40 ! KB volumsenter lekter
41 ! V volumdeplasement
42 !
43 ! SUBROUTINER:
44 ! Navn Funksjonsbeskrivelse
45 ! .....
46 ! konstanter beregner metasenter og andre arealmomenter
47 ! akalk beregner tilleggsmasse i hiv
48 ! bckalk beregner damping og fjærkoeffisienter
49 ! h3kalk beregner transferfunksjoner hiv
50 ! h5kalk beregner transferfunksjoner stamp
51 ! fasekalk beregner fasevinkel for bevegelser
52 !
53 ! LESEFILER:
54 ! Navn Eventuell beskrivelse
55 ! .....
56 ! konstantinput.txt inneholder lektergeometri
57 ! -----
58 !
59 ! Programmert av: Tor Erik Larsen
60 ! Dato/Versjon : 27.04.15 / 1.0
61 ! -----
62 !
63
64 !Starter subrutiner og erklærer alle variabler og konstanter
65 SUBROUTINE konstanter(L,Br,D,H,M,GML,I33,I55,rho,g,nl)
66 IMPLICIT NONE
67
68 !Erklærer variable og parametere, skriver kontroll til skjerm
69 INTEGER i,j,nl
70 REAL,DIMENSION(1:nl)::L,Br,D,H,GML,I33,I55,M,V
71 REAL,DIMENSION(1:nl)::KG,KB,BML
72 REAL::rho,g
73
74 !Leser inn lektergeometri fra input-fil, skriver ut kontroll til skjerm
75 OPEN(20,FILE="konstantinput.txt",ACTION="READ",STATUS="UNKNOWN")
76 DO j = 1,nl
77 READ(20,*) L(j),Br(j),D(j),H(j)
78 END DO

```



```

79 CLOSE(20)
80
81 !Beregner andre arealmoment for begge lektere, skriver ut kontroll
82 !Setter lekterfribord til dobbel dypgang
83 DO j=1,nl
84     V(j)    = L(j)*Br(j)*D(j)
85     M(j)    = rho*V(j)
86     I33(j)  = L(j) * Br(j)**3 / 12.0
87     I55(j)  = M(j) * ( (L(j)**2 / 12.0) + H(j)**2)
88     KG(j)   = D(j)
89     KB(j)   = D(j) / 2.0
90     BML(j)  = L(j)**2 / (12.0*D(j))
91     GML(j)  = KB(j) + BML(j) - KG(j)
92 END DO
93
94 !Gir tilbake kontroll og avslutter subrutine
95 RETURN
96 END SUBROUTINE

```

Listing 6: Subrutine akalk.f90

```

1  ! -----
2  !  DMET DELUX 2000 Subrutine akalk
3  ! -----
4  !  HENSIKT :
5  !  Beregner hydrodynamiske tilleggskoeffisienter for hiv
6  !
7  !  METODE :
8  !  Tar inn konstanter fra lektergeometri og frekvensvektor. Regner om
9  !  frekvens tilpasset avleste kurver i lærebok. Bruker innlagte
10 !  polynomer
11 !  til å beregne tilleggsmasse for todimensjonale striper av lekter.
12 !  Sender resultatmatriser for begge lektere videre til beregning av
13 !  transferfunksjoner. Skriver resultatfil som kan plottes ved kjøring
14 !  for visuell kontroll.
15 !
16 !  KALLESEKVENNS:
17 !  CALL akalk(n,nl,rho,g,omega,L,Br,D,A33,A55)
18 !
19 !  PARAMETERE:
20 !  Navn  I/O Type          Innhold/Beskrivelse
21 !  .....
22 !  omega I flyttalls vektor  bølgefrequens
23 !  n     I heltall          antall frekvensintervaller
24 !  nl    I/O heltall        antall lektere
25 !  k     I flyttalls vektor  bølgetall
26 !  zeta_an I flyttalls vektor bølgeamplitude
27 !  hs    I heltall          signifikant bølgehøyde
28 !  tp    I heltall          midlere periode
29 !  gamma I heltall          toppethetsfaktor
30 !  pi    I flyttall         pi
31 !  g     I/O flyttall        tyngdelfeltsakselerasjon
32 !  sn    0 flyttall         spekterverdi
33 !  a     I/O heltall        nedre frekvensverdi
34 !  b     I/O heltall        øvre frekvensverdi
35 !  H3    I/O flyttalls matrise transferfunksjon hiv

```

```

35 ! H5      I/O flyttalls matrise transferfunksjon stamp
36 ! delta3  I/O flyttalls matrise fasevinkel hiv
37 ! delta5  I/O flyttalls matrise fasevinkel stamp
38 ! A33     I/O flyttalls matrise tilleggsmasse hiv
39 ! A55     I/O flyttalls matrise tilleggsmasse stamp
40 ! B33     I/O flyttalls matrise demping hiv
41 ! B55     I/O flyttalls matrise demping stamp
42 ! C33     I/O flyttalls vektor  fjørkoeffisient hiv
43 ! C55     I/O flyttalls vektor  fjørkoeffisient stamp
44 ! L      I/O heltalls vektor   lengde på lekter
45 ! Br     I/O heltalls vektor   bredde på lekter
46 ! D      I/O flyttalls vektor  dypgang på lekter
47 ! H      I/O flyttalls vektor  dybde i riss for lekter
48 ! M      I/O flyttalls vektor  masse lekter
49 ! GML    I/O flyttalls vektor  langskips metasenter
50 ! I33    I/O flyttall vektor   andre arealmoment tverrskips
51 ! I55    I/O flyttalls vektor  andre arealmoment langskips
52 ! pi     I/O flyttall          pi
53 ! rho    I/O flyttall          tetthet sjøvann
54 ! g      I/O flyttall          tyngdeakselerasjonen
55 !
56 !
57 ! INTERNE VARIABLE:
58 ! Navn          Funksjon
59 !
60 ! .....
61 ! i             tellevariabler
62 ! oomegaA,oomegaB  omregning av frekvens
63 ! A33plot        plotteverdi avlest tilleggsmasse
64 !
65 ! SUBROUTINER:
66 ! Navn          Funksjonsbeskrivelse
67 !
68 ! .....
69 ! konstanter     beregner metasenter og andre arealmomenter
70 ! akalk          beregner tilleggsmasse i hiv
71 ! bckalk         beregner demping og fjørkoeffisienter
72 ! h3kalk         beregner transferfunksjoner hiv
73 ! h5kalk         beregner transferfunksjoner stamp
74 ! fasekalk       beregner fasevinkel for bevegelser
75 !
76 ! LESEFILER:
77 ! Navn          Eventuell beskrivelse
78 ! .....
79 ! konstantinput.txt inneholder lektergeometri
80 !
81 ! SKRIVEFILER:
82 ! Navn          Eventuell beskrivelse
83 !
84 ! .....
85 ! A33plot.txt    skriver resultat for visuell kontroll
86 !
87 ! -----

```

```

88 !
89 !   Programmert av:      Tor Erik Larsen
90 !   Dato/Versjon   :      27.04.15 / 1.0
91 !
92 !   -----
93
94 !Starter subrutine og erklærer alle variable og vektorer
95 SUBROUTINE akalk(n,nl,rho,g,omega,L,Br,D,A33,A55)
96 IMPLICIT NONE
97 INTEGER::i,n,nl
98 REAL::rho,g
99 DOUBLE PRECISION,DIMENSION(1:n)::omega,oomegaA,oomegaB
100 REAL,DIMENSION(1:nl)::L,Br,D,M,GML
101 REAL,DIMENSION(1:n,1:nl)::A33,A55,A33plot
102
103 !Beregner x-aksen for koeffisientene
104 DO i=1,n
105     oomegaA(i)=omega(i)*SQRT(Br(1)/(2.0*g))
106     oomegaB(i)=omega(i)*SQRT(Br(2)/(2.0*g))
107 END DO
108
109 !Beregner A33_2D for lekter A, sjettegradspolynom
110 DO i = 1,n
111     IF (oomegaA(i) < 2.0) THEN
112         A33plot(i,1) = ( &
113             -1.8169*oomegaA(i)**5 + 11.493*oomegaA(i)**4 -
114             28.060*oomegaA(i)**3 &
115             + 32.823*oomegaA(i)**2 - 17.894*oomegaA(i)**1 + 4.3851 &
116         )
117     ELSE
118         A33plot(i,1) = 1.15
119     END IF
120 END DO
121
122 !Beregner A33_2D for lekter B, femtegradspolynom
123 DO i = 1,n
124     IF (oomegaB(i) <2.0) THEN
125         A33plot(i,2) = (&
126             1.0534*oomegaB(i)**6 - 8.207*oomegaB(i)**5 &
127             + 26.398*oomegaB(i)**4 - 45.717*oomegaB(i)**3 &
128             + 46.022*oomegaB(i)**2 -25.507*oomegaB(i)**1 &
129             + 7.4189 &
130         )
131     ELSE
132         A33plot(i,2) = 1.9
133     END IF
134 END DO
135
136 !Beregner fullengde A33 ved å gange A33_2D med lengde
137 DO i=1,n
138     A33(i,1) = L(1) * A33plot(i,1)*(rho*Br(1)*D(1))
139     A33(i,2) = L(2) * A33plot(i,2)*(rho*Br(2)*D(2))
140 END DO
141
142 !Beregner A55 for begge lektere
143 DO i=1,n
144     A55(i,1) = (L(1)**2 / 12) * A33(i,1)

```

```

144     A55(i,2) = (L(2)**2 / 12) * A33(i,2)
145 END DO
146
147 !Bygger plottefil for å sjekke om koeffisientene er regnet riktig
148 OPEN(13,FILE="A33plot.txt",ACTION="WRITE",STATUS="UNKNOWN")
149 DO i=1,n
150     WRITE(13,*), oomegaA(i),A33plot(i,1),oomegaB(i), A33plot(i,2)
151 END DO
152 CLOSE(13)
153
154 !Gir tilbake kontrollen til hovedprogram og avslutter subrutine
155 RETURN
156 END SUBROUTINE

```

Listing 7: Subrutine bckalk.f90

```

1  ! -----
2  !  DMET DELUX 2000 Subrutine bckalk
3  !  -----
4  !  HENSIKT :
5  !  Beregner dempingskoeffisienter og fjærkoeffisienter
6  !
7  !  METODE :
8  !  Tar inn frekvensvektor og konstanter fra lektergeometri. Beregner
9  !  og sender videre dempings- og fjærkoeffisienter for hiv og stamp.
10 !
11 !  KALLESEKVENNS:
12 !  CALL bckalk(n,nl,rho,g,omega,L,Br,D,GML,B33,B55,C33,C55)
13 !
14 !  PARAMETERE:
15 !  Navn  I/O Type          Innhold/Beskrivelse
16 !  .....
17 !  omega I flyttalls vektor  bølgefrequens
18 !  n      I heltall          antall frekvensintervaller
19 !  nl     I/O heltall        antall lektere
20 !  k      I flyttallsvektor  bølgetall
21 !  zeta_an I flyttallsvektor  bølgeamplitude
22 !  hs     I heltall          signifikant bølgehøyde
23 !  tp     I heltall          midlere periode
24 !  gamma I heltall          toppethetsfaktor
25 !  pi     I flyttall         pi
26 !  g      I/O flyttall       tyngdelfeltsakselerasjon
27 !  sn     0 flyttallsvektor  spekterverdi
28 !  a      I/O heltall        nedre frekvensverdi
29 !  b      I/O heltall        øvre frekvensverdi
30 !  H3     I/O flyttalls matrise transferfunksjon hiv
31 !  H5     I/O flyttalls matrise transferfunksjon stamp
32 !  delta3 I/O flyttalls matrise fasevinkel hiv
33 !  delta5 I/O flyttalls matrise fasevinkel stamp
34 !  A33    I/O flyttalls matrise tilleggsmasse hiv
35 !  A55    I/O flyttalls matrise tilleggsmasse stamp
36 !  B33    I/O flyttalls matrise demping hiv
37 !  B55    I/O flyttalls matrise demping stamp
38 !  C33    I/O flyttalls vektor fjærkoeffisient hiv
39 !  C55    I/O flyttalls vektor fjærkoeffisient stamp
40 !  L      I/O heltalls vektor lengde på lekter

```

```

41 ! Br      I/O heltalls vektor  bredde på lekter
42 ! D      I/O flyttalls vektor  dypgang på lekter
43 ! H      I/O flyttalls vektor  dybde i riss for lekter
44 ! M      I/O flyttalls vektor  masse lekter
45 ! GML     I/O flyttalls vektor  langskips metasenter
46 ! I33     I/O flyttall vektor   andre arealmoment tverrskips
47 ! I55     I/O flyttalls vektor  andre arealmoment langskips
48 ! pi      I/O flyttall          pi
49 ! rho     I/O flyttall          tetthet sjøvann
50 ! g       I/O flyttall          tyngdeakselerasjonen
51 !
52 !
53 ! INTERNE VARIABLE:
54 ! Navn          Funksjon
55 !
56 ! .....
57 ! i,j           tellevariabler
58 ! oomegaA,oomegaB  omregning av frekvens
59 ! B33plot       plotteverdi avlest tilleggsmasse
60 !
61 ! SKRIVEFILER:
62 ! Navn          Eventuell beskrivelse
63 ! .....
64 ! B33plot.txt    skriver resultat for visuell kontroll
65 !
66 ! -----
67 !
68 ! Programmert av:      Tor Erik Larsen
69 ! Dato/Versjon   :      27.04.15 / 1.0
70 !
71 ! -----
72 !
73 !Starter subrutine og erklærer alle variable og vektorer
74 SUBROUTINE bckalk(n,nl,rho,g,omega,L,Br,D,GML,B33,B55,C33,C55)
75 IMPLICIT NONE
76 INTEGER::i,j,n,nl
77 REAL::rho,g
78 REAL,DIMENSION(1:nl)::L,Br,D,GML,C33,C55
79 REAL,DIMENSION(1:n,1:nl)::B33,B55,B33plot
80 DOUBLE PRECISION,DIMENSION(1:n)::omega,oomegaA,oomegaB
81
82 !Beregner x-aksen for koeffisientene
83 DO i=1,n
84     oomegaA(i) = omega(i)*SQRT(Br(1)/(2.0*g))
85     oomegaB(i) = omega(i)*SQRT(Br(2)/(2.0*g))
86 END DO
87
88 !Beregner B33_2D plotteverdi for lekter A, sjettegradspolynom
89 DO i=1,n
90     IF (oomegaA(i) < 1.47) THEN
91         B33plot(i,1) = ( &
92             0.985*oomegaA(i)**6 &
93             - 5.368*oomegaA(i)**5 + 10.474*oomegaA(i)**4 &
94             - 7.778*oomegaA(i)**3 + 0.077*oomegaA(i)**2 &
95             + 1.727*oomegaA(i)**1 + 0.004 &

```

```

96         )
97     ELSE
98         B33plot(i,1) = 0.001
99 END IF
100 END DO
101
102 !Beregner B33_2D plotteverdi for lekter B, femtegradspolynom
103 DO i=1,n
104     IF (oomegaB(i) < 2.03) THEN
105         B33plot(i,2) = ( &
106             - 0.220*oomegaB(i)**5 &
107             + 0.547*oomegaB(i)**4 + 1.431*oomegaB(i)**3 &
108             - 5.396*oomegaB(i)**2 + 4.212*oomegaB(i)**1 &
109             + 0.017 &
110         )
111     ELSE
112         B33plot(i,2) = 0.001
113     END IF
114 END DO
115
116 !Beregner fullengde B33, ved å gange A33_2D med lengde L
117 DO i=1,n
118     B33(i,1) = L(1) * B33plot(i,1) * (rho*Br(1)*D(1) * &
119         SQRT((2.0*g) / Br(1)) )
120     B33(i,2) = L(2) * B33plot(i,2) * (rho*Br(2)*D(2) * &
121         SQRT( (2.0*g) / Br(2) ) )
122 END DO
123
124 !Beregner koeffisientene i stamp, A55 og B55, UT FRA A33_2D
125 DO i=1,n
126     B55(i,1) = (L(1)**2 / 12) * B33(i,1)
127     B55(i,2) = (L(2)**2 / 12) * B33(i,2)
128 END DO
129
130 !Beregner fjærkoeffisientene C33 og C55
131 DO j=1,nl
132     C33(j) = rho*g*L(j)*Br(j)
133     C55(j) = rho*g*L(j)*Br(j)*D(j)*GML(j)
134 END DO
135
136 !Bygger plottefil for å sjekke at koeffisientene er regnet riktig
137 OPEN(44, FILE="B33plot.txt", ACTION="WRITE", STATUS="UNKNOWN")
138 DO i=1,n
139     WRITE(44,*) oomegaA(i), B33plot(i,1), oomegaB(i), B33plot(i,2)
140 END DO
141 CLOSE(44)
142
143 !Gir tilbake kontrollen til hovedprogram og avslutter subrutine
144 RETURN
145 END SUBROUTINE

```

Listing 8: Subrutine h3kalk.f90

```

1  ! -----
2  !  DMET DELUX 2000 Subrutine h3kalk
3  ! -----

```

```

4 ! HENSIKT :
5 ! Beregner transferfunksjoner i hiv for begge lektere
6 !
7 ! METODE :
8 ! Tar inn frekvensvektor, konstantverdier fra lektergeometri og
9 ! alle beregnede koeffisienter for hiv og stamp. Benytter
10 ! matematiske modeller for svingesystemer og kalkulerer
11 ! det dimensjonsløse utslaget for hivbevegelsen til begge lektere.
12 ! Teorien er presentert i rapporten. På grunn av komplekse formler
13 ! er uttrykkene faktorisert i stor grad.
14 !
15 ! KALLESEKVENST:
16 ! CALL h3kalk(k,omega,n,nl,rho,g,A33,B33,C33,L,Br,D,M,H3)
17 !
18 ! PARAMETERE:
19 ! Navn I/O Type Innhold/Beskrivelse
20 ! .....
21 ! omega I flyttalls vektor bølgefrequens
22 ! n I heltall antall frekvensintervaller
23 ! nl I/O heltall antall lektere
24 ! k I flyttallsvektor bølgetall
25 ! zeta_an I flyttallsvektor bølgeamplitude
26 ! hs I heltall signifikant bølgehøyde
27 ! tp I heltall midlere periode
28 ! gamma I heltall toppethetsfaktor
29 ! pi I flyttall pi
30 ! g I/O flyttall tyngdelfeltsakselerasjon
31 ! sn 0 flyttallsvektor spekterverdi
32 ! a I/O heltall nedre frekvensverdi
33 ! b I/O heltall øvre frekvensverdi
34 ! H3 I/O flyttalls matrise transferfunksjon hiv
35 ! H5 I/O flyttalls matrise transferfunksjon stamp
36 ! delta3 I/O flyttalls matrise fasevinkel hiv
37 ! delta5 I/O flyttalls matrise fasevinkel stamp
38 ! A33 I/O flyttalls matrise tilleggsmasse hiv
39 ! A55 I/O flyttalls matrise tilleggsmasse stamp
40 ! B33 I/O flyttalls matrise demping hiv
41 ! B55 I/O flyttalls matrise demping stamp
42 ! C33 I/O flyttalls vektor fjørkoeffisient hiv
43 ! C55 I/O flyttalls vektor fjørkoeffisient stamp
44 ! L I/O heltalls vektor lengde på lekter
45 ! Br I/O heltalls vektor bredde på lekter
46 ! D I/O flyttalls vektor dypgang på lekter
47 ! H I/O flyttalls vektor dybde i riss for lekter
48 ! M I/O flyttalls vektor masse lekter
49 ! GML I/O flyttalls vektor langskips metasenter
50 ! I33 I/O flyttall vektor andre arealmoment tverrskips
51 ! I55 I/O flyttalls vektor andre arealmoment langskips
52 ! pi I/O flyttall pi
53 ! rho I/O flyttall tetthet sjøvann
54 ! g I/O flyttall tyngdeakselerasjonen
55 !
56 !
57 ! INTERNE VARIABLE:
58 ! Navn Funksjon
59 !
.....

```

```

60 ! i               tellevariabler
61 ! F3A,F3A1,F3A2    faktorer i uttrykk
62 ! eta3nevner       faktor i uttrykk
63 ! eta3A            utslag for lekter i meter
64 !
65 !
66 ! SKRIVEFILER:
67 ! Navn             Eventuell beskrivelse
68 !
69 ! .....
69 ! H3plot.txt        skriver resultat for visuell kontroll
70 !
71 !
72 ! -----
73 !
74 ! Programmert av:    Tor Erik Larsen
75 ! Dato/Versjon   :    27.04.15 / 1.0
76 !
77 ! -----
78
79 SUBROUTINE h3kalk(k,omega,n,nl,rho,g,A33,B33,C33,L,Br,D,M,H3,zeta_an)
80 IMPLICIT NONE
81 !Erklærer variable og vektorer
82 INTEGER::n,nl,i,j
83 DOUBLE PRECISION,DIMENSION(1:n)::omega,k,zeta_an
84 REAL,DIMENSION(1:n,1:nl)::A33,B33
85 DOUBLE PRECISION,DIMENSION(1:n,1:nl)::eta3nevner,F3A1,F3A2,F3A
86 DOUBLE PRECISION,DIMENSION(1:n,1:nl)::eta3A,H3
87 REAL,DIMENSION(1:nl)::L,Br,D,M,C33
88 REAL rho,g,a,b
89
90 !Beregner integralene til kreftene i hiv, skriver kontroll til skjerm
91 DO i=1,n
92     DO j=1,nl
93         F3A1(i,j) = (2.0/k(i)) * SIN((k(i) * L(j)) / 2.0 )
94
95         F3A2(i,j) = rho*g*Br(j)* EXP(- k(i)*D(j) )&
96             - omega(i)**2 * ( A33(i,j)/L(j) ) * &
97             EXP(- k(i)*D(j) / 2.0)
98
99         eta3nevner(i,j)= SQRT((C33(j) - (M(j) + A33(i,j))*omega(i)**2
100             )**2 &
101             + (B33(i,j) * omega(i))**2 )
102
103         F3A (i,j) = F3A1(i,j) * F3A2(i,j)
104     END DO
105 END DO
106
107 !Beregner eta 3
108 DO i=1,n
109     DO j=1,nl
110         eta3A(i,j)= F3A(i,j) / eta3nevner(i,j)
111     END DO
112 END DO
113
114 !Beregner transfer-funksjoner
115 DO i=1,n

```



```

115     DO j=1,nl
116         H3(i,j)=ABS(eta3A(i,j))
117     END DO
118 END DO
119
120 OPEN(49,FILE="H3plot.txt",ACTION="WRITE",STATUS="UNKNOWN")
121 DO i=1,n
122     WRITE(49,*) omega(i),H3(i,1),H3(i,2)
123 END DO
124 CLOSE(49)
125
126 RETURN
127 END SUBROUTINE

```

Listing 9: Subrutine h5kalk.f90

```

1  ! -----
2  !  DMET DELUX 2000 Subrutine h5kalk
3  ! -----
4  !  HENSIKT :
5  !  Beregner transferfunksjoner i stamp for begge lektere
6  !
7  !  METODE :
8  !  Tar inn frekvensvektor, konstantverdier fra lektergeometri og
9  !  alle beregnede koeffisienter for stamp. Benytter
10 !  matematiske modeller for svingesystemer og kalkulerer
11 !  det dimensjonsløse utslaget for stampebevegelsen til begge lektere.
12 !  Teorien er presentert i rapporten. På grunn av komplekse formler
13 !  er uttrykkene faktorisert i stor grad.
14 !
15 !  KALLESEKVENSS:
16 !  CALL h5kalk(n,nl,L,Br,D,M,A33,B33,A55,B55,C55,I55,k,omega,rho,g,H5)
17 !
18 !  PARAMETERE:
19 !  Navn  I/O Type          Innhold/Beskrivelse
20 !  .....
21 !  omega I flyttalls vektor  bølgefrequens
22 !  n     I heltall          antall frekvensintervaller
23 !  nl    I/O heltall        antall lektere
24 !  k     I flyttallsvektor  bølgetall
25 !  pi    I flyttall         pi
26 !  g     I/O flyttall       tyngdelfeltsakselerasjon
27 !  H3    I/O flyttalls matrise transferfunksjon hiv
28 !  H5    I/O flyttalls matrise transferfunksjon stamp
29 !  delta3 I/O flyttalls matrise fasevinkel hiv
30 !  delta5 I/O flyttalls matrise fasevinkel stamp
31 !  A33   I/O flyttalls matrise tilleggsmasse hiv
32 !  A55   I/O flyttalls matrise tilleggsmasse stamp
33 !  B33   I/O flyttalls matrise demping hiv
34 !  B55   I/O flyttalls matrise demping stamp
35 !  C33   I/O flyttalls vektor fjørkoeffisient hiv
36 !  C55   I/O flyttalls vektor fjørkoeffisient stamp
37 !  L     I/O heltalls vektor lengde på lekter
38 !  Br    I/O heltalls vektor bredde på lekter
39 !  D     I/O flyttalls vektor dypgang på lekter
40 !  H     I/O flyttalls vektor dybde i riss for lekter

```

```

41 ! M I/O flyttalls vektor masse lekter
42 ! GML I/O flyttalls vektor langskips metasenter
43 ! I55 I/O flyttalls vektor andre arealmoment langskips
44 ! pi I/O flyttall pi
45 ! rho I/O flyttall tetthet sjøvann
46 ! g I/O flyttall tyngdeakselerasjonen
47 !
48 !
49 ! INTERNE VARIABLE:
50 ! Navn Funksjon
51 !
52 ! .....
53 ! i,j tellevariabler
54 ! F5A,F5A1,F5A2 faktorer i uttrykk
55 ! eta5nevner faktor i uttrykk
56 ! eta5A utslag for lekter i meter
57 !
58 ! SKRIVEFILER:
59 ! Navn Eventuell beskrivelse
60 !
61 ! .....
62 ! H5plot22.txt skriver resultat for visuell kontroll
63 !
64 ! -----
65 !
66 ! Programmert av: Tor Erik Larsen
67 ! Dato/Versjon : 27.04.15 / 1.0
68 !
69 ! -----
70
71 SUBROUTINE h5kalk(n,nl,L,Br,D,M,A33,B33,A55,B55 &
72 ,C55,I55,k,omega,rho,g,H5)
73 IMPLICIT NONE
74 !Erklærer variable
75 INTEGER::n,nl,i,j
76 REAL rho,g
77 REAL,DIMENSION(1:nl)::L,Br,D,M,I55,C55
78 REAL,DIMENSION(1:n,1:nl)::A33,B33,A55,B55
79 DOUBLE PRECISION,DIMENSION(1:n)::k,omega
80 DOUBLE PRECISION,DIMENSION(1:n,1:nl)::F5A1,F5A2,eta5nevner,F5A
81 DOUBLE PRECISION,DIMENSION(1:n,1:nl)::eta5A,H5
82
83 !Beregner eksitasjonskraft F5A
84 DO i=1,n
85 DO j=1,nl
86 F5A1(i,j) = (rho*g*Br(j)*EXP(-k(i)*D(j)) &
87 - omega(i)**2 * (A33(i,j)/L(j))* EXP(-k(i)*D(j)/2.0 ) )
88
89 F5A2(i,j) = L(j)*COS(k(i)*L(j) / 2.0) &
90 - (2.0 / k(i) )*SIN( k(i)*L(j) / 2.0 )
91
92 eta5nevner(i,j) = (k(i)**2) &
93 * SQRT( (C55(j) - (I55(j) + A55(i,j))*omega(i)**2 )**2 &
94 + (B55(i,j)*omega(i))**2 )
95 END DO

```

```

96  END DO
97
98  !Multipliserer faktorene i nevner
99  DO i=1,n
100     DO j=1,nl
101        F5A (i,j) = F5A1(i,j) * F5A2(i,j)
102     END DO
103  END DO
104
105  !Beregner eta5A
106  DO i=1,n
107     DO j=1,nl
108        eta5A(i,j) = F5A(i,j) / eta5nevner(i,j)
109     END DO
110  END DO
111
112  !Beregner transferfunksjonen H5
113  DO i=1,n
114     DO j=1,nl
115        H5(i,j) = ABS(eta5A(i,j))
116     END DO
117  END DO
118
119  !Skriver H5 til plottefil for visuell sjekk
120  OPEN(488,FILE="H5plot22.txt",ACTION="WRITE",STATUS="UNKNOWN")
121  DO i=1,n
122     WRITE(488,*) omega(i), H5(i,1), H5(i,2)
123  END DO
124  CLOSE(488)
125
126  RETURN
127  END SUBROUTINE h5kalk

```

Listing 10: Subrutine fase.f90

```

1  ! -----
2  !  DMET DELUX 2000 Subrutine fase
3  ! -----
4  !  HENSIKT :
5  !  Beregner fasevinkler for hiv og stamp for begge lektere
6  !
7  !  METODE :
8  !  Tar inn frekvensvektor, konstantverdier fra lektergeometri og
9  !  alle beregnede koeffisienter for stamp. Beregner fasevinkel
10 !  mellom krefter og respons ut fra teorien om dempede svingesystemer
11 !  med periodisk kraft. ATAN2-funksjonen i Fortran er brukt.
12 !  Teorien er presentert i rapporten. På grunn av komplekse formler
13 !  er uttrykkene faktorisert i stor grad.
14 !
15 !  KALLESEKVENNS:
16 !  CALL
17 !  fasekalk(n,nl,omega,M,A33,B33,C33,A55,B55,C55,I55,delta3,delta5)
18 !
19 !  PARAMETERE:
20 !  Navn  I/O Type          Innhold/Beskrivelse

```

```

21 ! omega I flyttalls vektor   bølgefrequens
22 ! n      I heltall           antall frekvensintervaller
23 ! n1     I/O heltall         antall lektere
24 ! k      I flyttallsvektor   bølgetall
25 ! zeta_an I flyttallsvektor   bølgeamplitude
26 ! hs     I heltall           signifikant bølgehøyde
27 ! tp     I heltall           midlere periode
28 ! gamma  I heltall           toppethetsfaktor
29 ! pi     I flyttall          pi
30 ! g      I/O flyttall        tyngdelfeltsakselerasjon
31 ! sn     0 flyttallsvektor   spekterverdi
32 ! a      I/O heltall         nedre frekvensverdi
33 ! b      I/O heltall         øvre frekvensverdi
34 ! H3     I/O flyttalls matrise transferfunksjon hiv
35 ! H5     I/O flyttalls matrise transferfunksjon stamp
36 ! delta3  I/O flyttalls matrise fasevinkel hiv
37 ! delta5  I/O flyttalls matrise fasevinkel stamp
38 ! A33    I/O flyttalls matrise tilleggsmasse hiv
39 ! A55    I/O flyttalls matrise tilleggsmasse stamp
40 ! B33    I/O flyttalls matrise demping hiv
41 ! B55    I/O flyttalls matrise demping stamp
42 ! C33    I/O flyttalls vektor fjørkoeffisient hiv
43 ! C55    I/O flyttalls vektor fjørkoeffisient stamp
44 ! L      I/O heltalls vektor lengde på lekter
45 ! Br     I/O heltalls vektor bredde på lekter
46 ! D      I/O flyttalls vektor dypgang på lekter
47 ! H      I/O flyttalls vektor dybde i riss for lekter
48 ! M      I/O flyttalls vektor masse lekter
49 ! GML    I/O flyttalls vektor langskips metasenter
50 ! I33    I/O flyttall vektor andre arealmoment tverrskips
51 ! I55    I/O flyttalls vektor andre arealmoment langskips
52 ! pi     I/O flyttall          pi
53 ! rho    I/O flyttall          tetthet sjøvann
54 ! g      I/O flyttall          tyngdeakselerasjonen
55 !
56 !
57 ! INTERNE VARIABLE:
58 ! Navn                      Funksjon
59 !
60 ! .....
61 ! i                          tellevariabler
62 ! t3,n3,t5,n5                faktorer i uttrykk!
63 !
64 ! SKRIVEFILER:
65 ! Navn                      Eventuell beskrivelse
66 ! .....
67 ! fase3sjekk.txt             skriver resultat for visuell kontroll
68 ! fase5sjekk.txt             skriver resultat for visuell kontroll
69 ! -----
70 !
71 ! Programmert av:            Tor Erik Larsen
72 ! Dato/Versjon :              27.04.15 / 1.0
73 !
74 ! -----
75 !

```

```

76 SUBROUTINE
   fasekalk(n,nl,omega,M,A33,B33,C33,A55,B55,C55,I55,delta3,delta5)
77 IMPLICIT NONE
78 !Erklærer vektorer og parametere
79 INTEGER::i,j,n,nl
80 REAL::rho,g
81 DOUBLE PRECISION::t3,n3,t5,n5
82 REAL,DIMENSION(1:n,1:nl)::A33,B33,A55,B55
83 REAL,DIMENSION(1:nl)::C33,C55,M,I55
84 DOUBLE PRECISION,DIMENSION(1:n)::omega
85 DOUBLE PRECISION,DIMENSION(1:n,1:nl)::delta3,delta5
86
87 !Regner fasevinkel for hivbevegelse
88 DO i=1,n
89     DO j=1,nl
90         t3 = (-omega(i)*B33(i,j))
91         n3 = (C33(j) - (M(j) + A33(i,j))*omega(i)**2)
92         delta3(i,j) = ATAN2(t3,n3)
93     END DO
94 END DO
95
96 !Regner fasevinkel for stampebevegelse
97 DO i=1,n
98     DO j=1,nl
99         t5 = omega(i)*B55(i,j)
100        n5 = (C55(j) - (I55(j) + A55(i,j))*omega(i)**2)
101        delta5(i,j) = ATAN2(t5,n5)
102    END DO
103 END DO
104
105 !Skriver hivfase til fil for visuell kontroll
106 OPEN(249,FILE="fase3sjekk.txt",ACTION="WRITE",STATUS="UNKNOWN")
107 DO i=1,n
108     WRITE(249,*)omega(i), delta3(i,1), delta3(i,2)
109 END DO
110 CLOSE(249)
111
112 !Skriver stampefasevinkel til fil for kontroll
113 OPEN(251,FILE="fase5sjekk.txt",ACTION="WRITE",STATUS="UNKNOWN")
114 DO i=1,n
115     WRITE(251,*) omega(i), delta5(i,1), delta5(i,2)
116 END DO
117 CLOSE(251)
118
119 RETURN
120 END SUBROUTINE

```

Listing 11: Program hoved.c

```

1 //
   -----
2 //   DMET DELUX 2000   Program   hoved
3 //
   -----
4 //   HENSIKT :
5 //   Definere vektorer, regne fasevinkel og kalle alle subrutiner som

```

```

6 //   behøves for å skrive visualiseringsfilen dmet.vtf
7 //
8 //   METODE :
9 //   Inkluderer header-fil og definerer samtlige konstanter og
    parametere
10 //   som programmet trenger. Kaller opp subrutinene suksessivt.
11 //
12 //   KALLESEKVENNS:
13 //   Kjører ved å skrive "dmet.sh" i terminal
14 //
15 //   PARAMETERE:
16 //   Navn I/O Type          Innhold/Beskrivelse
17 //
    .....
18 //   omega  I flyttalls vektor   bølgefrequens
19 //   n      I heltall            antall frekvensintervaller
20 //   nl     I/O heltall          antall lektere
21 //   k      I flyttallsvektor    bølgetall
22 //   A      I flyttallsvektor    bølgeamplitude
23 //   pi     I flyttall           pi
24 //   g      I/O flyttall         tyngdelfeltsakselerasjon
25 //   a      I/O flyttall         2pi
26 //   H3     I/O flyttalls matrise transferfunksjon hiv
27 //   H5     I/O flyttalls matrise transferfunksjon stamp
28 //   delta3 I/O flyttalls matrise fasevinkel hiv
29 //   delta5 I/O flyttalls matrise fasevinkel stamp
30 //   L      I/O heltalls vektor  lengde på lekter
31 //   B      I/O heltalls vektor  bredde på lekter
32 //   D      I/O flyttalls vektor dypgang på lekter
33 //   H      I/O flyttalls vektor dybde i riss for lekter
34 //   pi     I/O flyttall         pi
35 //   rho    I/O flyttall         tetthet sjøvann
36 //   g      I/O flyttall         tyngdeakselerasjonen
37 //   epsilon I/O flyttall       fasevinkel respons
38 //
39 //   INTERNE VARIABLE:
40 //   Navn          Funksjon
41 //
    .....
42 //   i,j          tellevariabler
43 //
44 //   SUBROUTINER:
45 //   Navn          Funksjonsbeskrivelse
46 //
    .....
47 //   lesInn        leser inn resultatfil fra Fortran-program
48 //   glModell       bygger første del av VTF-fil med geometri
49 //   havFlate       skriver overflateheving til havteppe
50 //   respons        skriver respons for lektere til VTF-fil
51 //   kontroll       sammenstiller resultater i tidssteg
52 //
53 //
    -----
54 //
55 //   Programmert av:      Tor Erik Larsen
56 //   Dato/Versjon   :      27.04.15 / 1.0
57 //

```

```

58 //
59 -----
60 #include "header.h"
61
62 //Definerer hovedfil
63 int main(){
64     int i,j;
65     int L[NL];
66     int B[NL];
67     float D[NL],H[NL];
68     double omega[N],k[N],A[N];
69     double H3[N][NL],H5[N][NL],delta3[N][NL],delta5[N][NL];
70     double epsilon[N],a=2.0*pi;
71     //srand(time(NULL));
72
73     //Beregner fasevinkel for å sende til hav.c og respons
74     for (j=0;j<N;j++){
75         epsilon[j] = ((double)rand()/((double)(RAND_MAX)))*a;
76     }
77     //Kaller opp alle subrutiner
78     lesInn(L,B,D,H,omega,k,A,H3,H5,delta3,delta5);
79     glModell(L,B,D,H);
80     havFlate(A,omega,k,epsilon);
81     respons(L,B,D,H,omega,k,A,H3,H5,delta3,delta5,epsilon);
82     kontroll();
83
84     return 0;
85 }

```

Listing 12: Subrutine lesinn.c

```

1 //
2 // -----
3 //
4 // HENSIKT :
5 // Lese inn resultater fra Fortran-programmet og bygge vektorer for
6 // bruk videre i programmet.
7 //
8 // METODE :
9 // Opne tekstfilen fra Fortran-programmet, og benytte fscanf til å
10 // lese
11 // tekstfil linje for linje. For hver linje legges verdiene inn i
12 // forhåndsdefinerte vektorer. I tillegg leses geometrifik for
13 // lektere.
14 //
15 // KALLESEKVENSS:
16 // lesInn(L,B,D,H,omega,k,A,H3,H5,delta3,delta5);
17 //
18 // PARAMETERE:
19 // Navn I/O Type Innhold/Beskrivelse
20 //
21 // .....
22 // omega I/O flyttals vektor bølgefrequens

```

```

20 //      n      I heltall          antall frekvensintervaller
21 //      nl     I heltall          antall lektere
22 //      k      I/O flyttallsvektor  bølgetall
23 //      A      I/O flyttallsvektor  bølgeamplitude
24 //      pi     I flyttall          pi
25 //      g      I/O flyttall          tyngdelfeltsakselerasjon
26 //      a      I/O flyttall          2pi
27 //      H3     I/O flyttalls matrise transferfunksjon hiv
28 //      H5     I/O flyttalls matrise transferfunksjon stamp
29 //      delta3 I/O flyttalls matrise fasevinkel hiv
30 //      delta5 I/O flyttalls matrise fasevinkel stamp
31 //      L      I/O heltalls vektor  lengde på lekter
32 //      B      I/O heltalls vektor  bredde på lekter
33 //      D      I/O flyttalls vektor  dypgang på lekter
34 //      H      I/O flyttalls vektor  dybde i riss for lekter
35 //
36 //      INTERNE VARIABLE:
37 //      Navn      Funksjon
38 //
39 //      .....
40 //      i,j      tellevariabler
41 //
42 //      SUBROUTINER:
43 //      Navn      Funksjonsbeskrivelse
44 //
45 //      .....
46 //      lesInn      leser inn resultatfil fra Fortran-program
47 //      glModell    bygger første del av VTF-fil med geometri
48 //      havFlate    skriver overflateheving til havteppe
49 //      respons     skriver respons for lektere til VTF-fil
50 //      kontroll    sammenstiller resultater i tidssteg
51 //
52 //      LESEFILER:
53 //      Navn      Eventuell beskrivelse
54 //      .....
55 //      konstantinput.txt  inneholder lektergeometri
56 //      fortranresultat.txt  inneholder resultater fra Fortran-prog
57 //
58 //      -----
59 //      Programmert av:      Tor Erik Larsen
60 //      Dato/Versjon   :      27.04.15 / 1.0
61 //
62 //      -----
63
64 #include "header.h"
65
66 //Definerer subrutine som kjøres i denne filen
67
68 void lesInn(int L[NL],int B[NL],float D[NL],float H[NL], \
69            double omega[N],double k[N],double A[N], \
70            double H3[N][NL],double H5[N][NL], \
71            double delta3[N][NL],double delta5[N][NL]){

```



```

72
73 //Definerer filnavn og erklærer telletall
74 FILE * filpeker;
75 int i,j;
76
77 //Laster inn konstantfil og fyller lektergeometri i vektorene
78 filpeker = fopen("konstantinput.txt", "r");
79 for(i=0;i<NL;i++){
80     fscanf(filpeker,"%d %d %f %f",&L[i],&B[i],&D[i],&H[i]);
81 }
82 fclose(filpeker);
83
84 //Laster inn resultatfil og fyller vektorer
85 filpeker = fopen("fortranresultat.txt", "r");
86 for(i=0;i<N;i++){
87     fscanf(filpeker,"%lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf", \
88         &omega[i],&k[i],&A[i],&H3[i][0],&H3[i][1],&H5[i][0],&H5[i][1], \
89         &delta3[i][0],&delta3[i][1],&delta5[i][0],&delta5[i][1]);
90 }
91 fclose(filpeker);
92
93 //Gir tilbake kontroll og avslutter subrutine
94 return;
95 }

```

Listing 13: Subrutine glmodell.c

```

1 //
2 // -----
3 //
4 // DMET DELUX 2000 Subrutine glmodell
5 //
6 // -----
7 //
8 // HENSIKT :
9 // Starte skriving til VTF-fil for visualisering i GLView. Definerer
10 // alle noder og elementer for både havteppe og lektere
11 //
12 // METODE :
13 // Ta inn havteppegeometri fra header-fil, og ta inn lektergeometri
14 // fra tidligere subrutine. Skrive filhode og starte noder og
15 // elementer.
16 //
17 // KALLESEKVENSS:
18 // glModell(L,B,D,H);
19 //
20 // PARAMETERE:
21 // Navn I/O Type Innhold/Beskrivelse
22 //
23 // .....
24 // n I heltall antall frekvensintervaller
25 // n1 I heltall antall lektere
26 // L I heltalls vektor lengde på lekter
27 // B I heltalls vektor bredde på lekter
28 // D I flyttalls vektor dypgang på lekter
29 // H I flyttalls vektor dybde i riss for lekter
30 //
31 // INTERNE VARIABLE:

```

```

26 //      Navn                      Funksjon
27 //
28 //      .....
29 //      i,j                      tellevariabler
30 //      nodegruppe,x,xkoordinat  nodevariabler
31 //      xsenter,ysenter          definerer senter i havteppe
32 //      senterlinje              definerer senterlinjer på lektene
33 //
34 //      SKRIVEFILER:
35 //      Navn                      Eventuell beskrivelse
36 //
37 //      dmet.vtf                  Visualiseringsfil for GLView
38 //
39 //
40 //
41 //      Programmert av:           Tor Erik Larsen
42 //      Dato/Versjon   :           27.04.15 / 1.0
43 //
44 //
45
46 #include "header.h"
47
48 void glModell(int L[NL],int B[NL],float D[NL],float H[NL]){
49     int i,j,nodegruppe,n,xstart;
50     double senterlinje[2];
51     double xkoordinat,xsenter,ysenter,ykoordinat,x;
52     FILE * filpeker;
53
54     //Åpner VTF-fil og starter skriving
55     filpeker = fopen("dmet.vtf", "w");
56     fprintf(filpeker,"%s\n","*VTF-1.00");
57     fprintf(filpeker,"%s\n","*NODES 1");
58     fprintf(filpeker,"%s\n","*WITH_ID");
59     n=1;
60     xstart = -havlengde/2;
61     ykoordinat = havbredde/2.0;
62     for(i=xstart;i<havlengde/2;i++){
63         x = i*1.0;
64         xkoordinat = i+0.5;
65         fprintf(filpeker,"%d \t %.1f %.1f %.1f\n",n,x,-ykoordinat,0.0);
66         fprintf(filpeker,"%d \t %.1f %.1f
67             %.1f\n",n+1,xkoordinat,-ykoordinat,0.0);
68         fprintf(filpeker,"%d \t %.1f %.1f
69             %.1f\n",n+2,xkoordinat,ykoordinat,0.0);
70         fprintf(filpeker,"%d \t %.1f %.1f %.1f\n",n+3,x,ykoordinat,0.0);
71         n = n+4;
72     }
73
74     //Definerer elementer for havflaten ut fra nodene ovenfor
75     fprintf(filpeker,"%s\n","*ELEMENTS 1");
76     fprintf(filpeker,"%s\n","*COLORS 0,0,1");
77     fprintf(filpeker,"%s\n","*NODES #1");
78     fprintf(filpeker,"%s\n","*QUADS");

```

```

77
78 n=1;
79 for(x=0;x<havlengde-1;x++){
80     fprintf(filpeker,"%d %d %d %d\n",n,n + 1,n + 2,n + 3);
81     fprintf(filpeker,"%d %d %d %d\n",n + 1,n + 4,n + 7,n + 2);
82     n = n +4;
83 }
84 n=0;
85 //Bygger lekternoder
86 senterlinje[0] = 1.0 + B[0]/2.0;
87 senterlinje[1] = - 1.0 - B[1]/2.0;
88 for(j=0;j<NL;j++){
89     nodegruppe = j+2;
90     n= nodegruppe*1000;
91     fprintf(filpeker,"%s %d \n","*NODES ",nodegruppe);
92     fprintf(filpeker,"%s \n","%WITH_ID");
93     fprintf(filpeker,"%d %.1f %.1f %f\n",n+1,-L[j]/2.0,senterlinje[j] \
94         -B[j]/2.0,-H[j]/2.0);
95     fprintf(filpeker,"%d %.1f %.1f %f\n",n+2,L[j]/2.0,senterlinje[j] \
96         -B[j]/2.0,-H[j]/2.0);
97     fprintf(filpeker,"%d %.1f %.1f %f\n",n+3,L[j]/2.0,senterlinje[j]+\
98         B[j]/2.0,-H[j]/2.0);
99     fprintf(filpeker,"%d %.1f %.1f %f\n",n+4,-L[j]/2.0,senterlinje[j]+\
100         B[j]/2.0,-H[j]/2.0);
101     fprintf(filpeker,"%d %.1f %.1f %f\n",n+5,-L[j]/2.0,senterlinje[j]-\
102         B[j]/2.0,H[j]/2.0);
103     fprintf(filpeker,"%d %.1f %.1f %f\n",n+6,L[j]/2.0,senterlinje[j]-\
104         B[j]/2.0,H[j]/2.0);
105     fprintf(filpeker,"%d %.1f %.1f %f\n",n+7,L[j]/2.0,senterlinje[j]+\
106         B[j]/2.0,H[j]/2.0);
107     fprintf(filpeker,"%d %.1f %.1f %f\n",n+8,-L[j]/2.0,senterlinje[j]+\
108         B[j]/2.0,H[j]/2.0);
109     fprintf(filpeker,"%s%d\n","*ELEMENTS ",nodegruppe);
110     fprintf(filpeker,"%s%d\n","%NODES #",nodegruppe);
111     fprintf(filpeker,"%s\n","%HEXAHEDRONS");
112     fprintf(filpeker,"%d %d %d %d %d %d %d %d\n",n+1,n+2,n+3,n+4,n+5,n+6,n+7,n+8);
113 }
114 fclose(filpeker);
115 return;
116 }

```

Listing 14: Subrutine respons.c

```

1 //
2 // -----
3 //   DMET DELUX 2000   Subrutine respons
4 //   -----
5 //   HENSIKT :
6 //   Skrive utslag for lekternoder som følge av bølger til VTF-fil
7 //   METODE :
8 //   Ta inn transferfunksjoner fra Fortran-program, beregne respons fra
9 //   responsligning, beregne utslag i hver enkelt lekternode og skrive
10 //   fysisk utslag til VTF-fil

```

```

11 //
12 //   KALLESEKVENST:
13 //   respons(L,B,D,H,omega,k,A,H3,H5,delta3,delta5,epsilon);
14 //
15 //   PARAMETERE:
16 //   Navn I/O Type           Innhold/Beskrivelse
17 //
18 //   .....
19 //   omega  I flyttalls vektor   bølgefrequens
20 //   n      I heltall            antall frekvensintervaller
21 //   n1     I heltall            antall lektere
22 //   k      I flyttallsvektor    bølgetall
23 //   A      I flyttallsvektor    bølgeamplitude
24 //   pi     I flyttall           pi
25 //   g      I flyttall           tyngdelfeltsakselerasjon
26 //   a      I flyttall           2pi
27 //   H3     I flyttalls matrise  transferfunksjon hiv
28 //   H5     I flyttalls matrise  transferfunksjon stamp
29 //   delta3 I flyttalls matrise  fasevinkel hiv
30 //   delta5 I flyttalls matrise  fasevinkel stamp
31 //   L      I heltalls vektor    lengde på lekter
32 //   B      I heltalls vektor    bredde på lekter
33 //   D      I flyttalls vektor    dypgang på lekter
34 //   H      I flyttalls vektor    dybde i riss for lekter
35 //   pi     I flyttall           pi
36 //   rho    I flyttall           tetthet sjøvann
37 //   g      I flyttall           tyngdeakselerasjonen
38 //   epsilon I flyttall          fasevinkel respons
39 //   fps    I heltall            steg per sekund
40 //   sx1    0 flyttall           utslag i x-retning nodepar 1
41 //   sx2    0 flyttall           utslag i x-retning nodepar 2
42 //   sx5    0 flyttall           utslag i x-retning nodepar 5
43 //   sx6    0 flyttall           utslag i x-retning nodepar 6
44 //   sz1    0 flyttall           utslag i z-retning nodepar 1
45 //   sz2    0 flyttall           utslag i z-retning nodepar 2
46 //   sz5    0 flyttall           utslag i z-retning nodepar 5
47 //   sz6    0 flyttall           utslag i z-retning nodepar 6
48 //
49 //   INTERNE VARIABLE:
50 //   Navn           Funksjon
51 //   .....
52 //   i,j           tellevariabler
53 //   lekter,teller  tellevariabler
54 //   xkoordinat     koordinatvariabel
55 //   Zbunn,Zdekk    koordinater z-akse
56 //   xBaug,xDekk    xkoordinater for noder
57 //   s1,s2,5,s6     utslagsvariabler
58 //   xnull,znull    nullpunkter lektere
59 //   t              tid
60 //
61 //   SUBROUTINER:
62 //   Navn           Funksjonsbeskrivelse
63 //   .....
64 //   lesInn         leser inn resultatfil fra Fortran-program
65 //   glModell       bygger første del av VTF-fil med geometri

```

```

65 //   havFlate           skriver overflateheving til havteppe
66 //   respons           skriver respons for lektere til VTF-fil
67 //   kontroll          sammenstiller resultater i tidssteg
68 //
69 //   LESEFILER:
70 //   Navn               Eventuell beskrivelse
71 //
72 //   .....
73 //   konstantinput.txt   Inneholder lektergeometri
74 //   fortranresultat.txt Inneholder resultater fra Fortran-prog
75 //
76 //   SKRIVEFILER:
77 //   Navn               Eventuell beskrivelse
78 //   .....
79 //
80 //
81 //
82 //   Programmert av:      Tor Erik Larsen
83 //   Dato/Versjon   :      27.04.15 / 1.0
84 //
85 //
86
87 #include "header.h"
88
89 void respons(int L[NL],int B[NL],float D[NL],float H[NL], \
90             double omega[N],double k[N],double A[N], \
91             double H3[N][NL],double H5[N][NL], \
92             double delta3[N][NL],double delta5[N][NL],\
93             double epsilon[N]){
94
95     //Definerer filnavn og erklærer telletall
96     FILE * filpeker;
97     FILE * filpeker2;
98     int i,j,m,n,lekter,teller;
99     double xkoordinat,t;
100    double zBunn,zDekk,xBaug,xHekk,xnull,znull;
101    double s1,s2,s5,s6,eta3,eta5,eta3sjekk,eta5sjekk;
102    double sx1,sx2,sx5,sx6;
103    double sz1,sz2,sz5,sz6;
104    //srand(time(NULL));
105
106    //Åpner VTF-fil og starter skriving
107    filpeker = fopen("dmet.vtf", "a");
108    filpeker2 = fopen("responsplott.dat","w");
109    //Beregner respons for hiv, som skal gjelde alle noder
110    for(m=0;m<NL;m++){
111        for(i=0;i<steg;i++){
112            lekter = m+1;
113            teller = steg*lekter+i;
114            fprintf(filpeker,"*RESULTS %d\n",teller);
115            fprintf(filpeker,"%s","%DIMENSION 3\n");
116            fprintf(filpeker,"%s%d\n","%PER_NODE #",lekter+1);
117            t=i*1.0/fps;

```

```

118     eta3 = 0;
119     eta5 = 0;
120     eta3sjekk = 0;
121     eta5sjekk = 0;
122     for(j=0;j<N;j+=intervall){
123     eta3 = eta3 + A[j]*H3[j][m]*sin(omega[j]*t + delta3[j][m] +
124     epsilon[j]);
125     eta5 = eta5 + A[j]*H5[j][m]*k[j]*cos(omega[j]*t + delta5[j][m] +
126     epsilon[j]);
127     }
128     for(j=133;j<134;j++){
129     eta3sjekk = eta3sjekk + A[j]*H3[j][m]*sin(omega[j]*t + delta3[j][m]\
130     + epsilon[j]);
131     eta5sjekk = eta5sjekk + A[j]*H5[j][m]*k[j]*cos(omega[j]*t +
132     delta5[j][m] \
133     + epsilon[j]);
134     }
135     fprintf(filpeker2,"%lf %lf\n",eta3sjekk,eta5sjekk);
136     //Beregner respons fra bevegesligning
137     //printf("Lengde %d, bredde %d, høgde%f\n",L[m],B[m],H[m]);
138     zBunn = -H[m]/2.0;
139     zDekk = H[m]/2.0;
140     xBaug = -L[m]/2.0;
141     xHekk = L[m]/2.0;
142     sx1 = zBunn*eta5;
143     sz1 = eta3 -xBaug*eta5;
144     sx2 = zBunn*eta5;
145     sz2 = eta3 -xHekk*eta5;
146     sx5 = zDekk*eta5;
147     sz5 = eta3 -xBaug*eta5;
148     sx6 = zDekk*eta5;
149     sz6 = eta3 -xHekk*eta5;
150     fprintf(filpeker,"%f %d %f\n", sx1, 0, sz1);
151     fprintf(filpeker,"%f %d %f\n", sx2, 0, sz2);
152     fprintf(filpeker,"%f %d %f\n", sx2, 0, sz2);
153     fprintf(filpeker,"%f %d %f\n", sx1, 0, sz1);
154     fprintf(filpeker,"%f %d %f\n", sx5, 0, sz5);
155     fprintf(filpeker,"%f %d %f\n", sx6, 0, sz6);
156     fprintf(filpeker,"%f %d %f\n", sx6, 0, sz6);
157     fprintf(filpeker,"%f %d %f\n", sx5, 0, sz5);
158     }
159     fclose(filpeker);
160     fclose(filpeker2);
161     return;
162 }

```

Listing 15: Subrutine hav.c

```

1  //
2  //  -----
3  //  DMET DELUX 2000  Subrutine hav
4  //  -----
5  //  HENSIKT :

```

```

5 // Skrive overflateheving for havteppenoder til VTF-fil
6 //
7 // METODE :
8 // Ta inn frekvensvektor, fasevinkel, bølgeamplitude og bølgetall.
9 // Beregne overflateheving for hver enkelt node i havteppet, og skrive
10 // resultatet til VTF-filen.
11 //
12 // KALLESEKVENNS:
13 // havFlate(A,omega,k,epsilon);
14 //
15 // PARAMETERE:
16 // Navn I/O Type          Innhold/Beskrivelse
17 //
18 // .....
19 // omega I flyttalls vektor   bølgefrekvens
20 // n      I heltall           antall frekvensintervaller
21 // n1     I heltall           antall lektere
22 // k      I flyttallsvektor   bølgetall
23 // A      I flyttallsvektor   bølgeamplitude
24 // pi     I flyttall          pi
25 // g      I flyttall          tyngdelfeltsakselerasjon
26 // a      I flyttall          2pi
27 // H3     I flyttalls matrise transferfunksjon hiv
28 // H5     I flyttalls matrise transferfunksjon stamp
29 // delta3 I flyttalls matrise fasevinkel hiv
30 // delta5 I flyttalls matrise fasevinkel stamp
31 // L      I heltalls vektor   lengde på lekter
32 // B      I heltalls vektor   bredde på lekter
33 // D      I flyttalls vektor   dypgang på lekter
34 // H      I flyttalls vektor   dybde i riss for lekter
35 // pi     I flyttall          pi
36 // rho    I flyttall          tetthet sjøvann
37 // g      I flyttall          tyngdeakselerasjonen
38 // epsilon I flyttall        fasevinkel respons
39 // fps    I heltall          steg per sekund
40 // Zeta 0  I flyttall          utslag i meter for noder
41 //
42 // INTERNE VARIABLE:
43 // Navn          Funksjon
44 //
45 // .....
46 // i,j           tellevariabler
47 // t             tid
48 // Zetasjekk     overflateheving for plottefil
49 // x, halvx      mellomvariabler for utslag
50 //
51 // SUBROUTINER:
52 // Navn          Funksjonsbeskrivelse
53 // .....
54 // lesInn        leser inn resultatfil fra Fortran-program
55 // glModell      bygger første del av VTF-fil med geometri
56 // havFlate      skriver overflateheving til havteppe
57 // respons       skriver respons for lektere til VTF-fil
58 // kontroll      sammenstiller resultater i tidssteg

```

```

59 //
60 //   SKRIVEFILER:
61 //   Navn           Eventuell beskrivelse
62 //
63 //   .....
64 //   dmet.vtf         visualiseringsfil
65 //   overflateplott.dat   overflatehevingsverdier for x
66 //
67 //   -----
68 //   Programmert av:      Tor Erik Larsen
69 //   Dato/Versjon   :      27.04.15 / 1.0
70 //
71 //   -----
72
73 #include "header.h"
74
75 void havFlate(double A[N], double omega[N], double k[N],double
    epsilon[N]){
76     int i,j,n,x,xstart;
77     double Zeta[2],Zetasjekk,t;
78     double halvx;
79     FILE * filpeker;
80     FILE * filpeker2;
81     //srand (time(NULL));
82
83     //Åpner VTF-fil og starter skriving
84     filpeker = fopen("dmet.vtf", "a");
85     filpeker2 = fopen("overflateplott.dat","w");
86     xstart = havlengde/2.0;
87     for(i=0;i<steg;i++){
88         fprintf(filpeker,"*RESULTS %d\n",i+1);
89         fprintf(filpeker,"%s","%DIMENSION 3\n");
90         fprintf(filpeker,"%s","%PER_NODE #1\n");
91         t= i*1.0/fps;
92         Zetasjekk = 0;
93         for(j=133;j<134;j++){
94             Zetasjekk = Zetasjekk + A[j]*sin(omega[j]*t - k[j]*(0) \
95                 + epsilon[j]);
96         }
97         for(n=-125;n<125;n++){
98             Zeta[0] = 0;
99             Zeta[1] = 0;
100             x = n*1.0;
101             halvx = n+0.5;
102             for(j=0;j<N;j+=intervall){
103                 Zeta[0] = Zeta[0] + A[j]*sin(omega[j]*t - k[j]*x + epsilon[j]);
104                 Zeta[1] = Zeta[1] + A[j]*sin(omega[j]*t - k[j]*halvx + epsilon[j]);
105             }
106             fprintf(filpeker,"%d %d %f\n", 0, 0, Zeta[0]);
107             fprintf(filpeker,"%d %d %f\n", 0, 0, Zeta[1]);
108             fprintf(filpeker,"%d %d %f\n", 0, 0, Zeta[1]);
109             fprintf(filpeker,"%d %d %f\n", 0, 0, Zeta[0]);
110         }
111         fprintf(filpeker2,"%lf\n",Zetasjekk);

```



```

112 }
113 fclose(filpeker);
114 fclose(filpeker2);
115 //Gir tilbake kontroll til program
116 return;
117 }

```

Listing 16: Subrutine kontroll.c

```

1 //
2 //  -----
3 //  DMET DELUX 2000  Subrutine kontroll
4 //  -----
5 //  HENSIKT :
6 //  Sammenstille resultater i VTF-fil, samordne tidssteg
7 //  METODE :
8 //  opne VTF-fil, og skrive en såkalt GLVIEWVECTOR til fil. Denne
9 //  samordner hvert enkelt resultat for hvert enkelt tidssteg, slik
10 //  at visualiseringen henger sammen
11 //
12 //  KALLESEKVENSS:
13 //  kontroll();
14 //
15 //  PARAMETERE:
16 //  Navn I/O Type      Innhold/Beskrivelse
17 //
18 //  ....
19 //  tid  0 flyttall      tid for hvert steg
20 //  fps  1 flyttall      tidsintervall
21 //  i    0 heltall       resultatteller
22 //
23 //  INTERNE VARIABLE:
24 //  Navn      Funksjon
25 //
26 //  ....
27 //
28 //  SKRIVEFILER:
29 //  Navn      Eventuell beskrivelse
30 //
31 //  ....
32 //  dmet.vtf      visualiseringsfil
33 //
34 //  -----
35 //  Programmert av:      Tor Erik Larsen
36 //  Dato/Versjon   :      27.04.15 / 1.0
37 //
38 //  -----
39
40 #include "header.h"

```

```

41
42 void kontroll(){
43     int i;
44     double tid;
45
46     FILE * filpeker;
47
48     //opner VTF-fil og starter skriving
49     filpeker = fopen("dmet.vtf", "a");
50
51     //Definerer GLVIEW-geometri
52     fprintf(filpeker, "*GLVIEWGEOMETRY 1\n");
53     fprintf(filpeker, "%s", "%ELEMENTS\n");
54     fprintf(filpeker, "1,2,3\n");
55
56     //Definerer GLVIEW-vektor, som samordner resultater
57     fprintf(filpeker, "*GLVIEWVECTOR 1\n");
58     fprintf(filpeker, "%s\n", "%NAME \"Displacements\"");
59     for(i=1; i<steg; i++){
60         tid = i*1.0/fps;
61         fprintf(filpeker, "%s %d\n", "%STEP", i);
62         fprintf(filpeker, "%s %.1f%s\n", "%STEPNAME \"Tid: ", tid, "s\"");
63         fprintf(filpeker, "%d,%d,%d\n", i, i+steg, i+steg*2);
64     }
65
66     fclose(filpeker);
67     return;
68 }

```

Listing 17: header.h

```

1 //
2 // -----
3 //   DMET DELUX 2000   header
4 // -----
5 //   HENSIKT :
6 //   Inkludere innebygde C-biblioteker, definerere konstanter og erklære
7 //   subrutiner i programmet
8 //   METODE :
9 //
10 //   KALLESEKVENNS:
11 //   #include "header.h"
12 //
13 //   PARAMETERE:
14 //   Navn I/O Type           Innhold/Beskrivelse
15 //   -----
16 //   pi      0 flyttall      pi
17 //   intervall 0 heltall      steg mellom bølger
18 //   N       0 heltall      antall bølger
19 //   NL      0 heltall      antall lektere
20 //   fps     0 heltall      tidsintervall per sekund
21 //   steg    0 heltall      antall resultater
22 //   havlengde 0 heltall      lengde havteppe

```

```

23 //   havbredde  0 heltall      bredde havteppe
24 //   stripebredde 0 heltall      bredde havstripe
25 //   nstriper   0 heltall      antall havstriper totalt
26 //
27 //   SUBROUTINER
28 //   Navn                      Eventuell beskrivelse
29 //
30 //   .....
31 //   lesInn                      Se beskrivelse i hoved.c
32 //   glModell
33 //   havFlate
34 //   respons
35 //   kontroll
36 //
37 //
38 //   Programmert av:      Tor Erik Larsen
39 //   Dato/Versjon   :      27.04.15 / 1.0
40 //
41 //
42 //   -----
43 #include <stdio.h>
44 #include <math.h>
45 #include <stdlib.h>
46 #include <time.h>
47
48 #define pi 3.1415
49
50 //Definerer antall bølgekomponenter og antall lektere
51 #define intervall 1
52 #define N 1000
53 #define NL 2
54
55 //Definerer "lukkertid" og totalt antall steg
56 #define fps 4
57 #define steg 600
58
59 //Definerer havgeometri og inndeling for visualisering
60 #define havlengde 250
61 #define havbredde 100.0
62 #define stripebredde 0.5
63 #define nstriper 300
64
65 //Erklærer subrutiner som brukes i programmet
66 void lesInn(int L[NL],int B[NL],float D[NL],float H[NL], \
67            double omega[N],double k[N],double A[N], \
68            double H3[N][NL],double H5[N][NL], \
69            double delta3[N][NL],double delta5[N][NL]);
70
71 void glModell(int L[NL],int B[NL],float D[NL],float H[NL]);
72
73 void havFlate(double A[N], double omega[N], double k[N],\
74             double epsilon[N]);
75
76 void respons(int L[NL],int B[NL],float D[NL],float H[NL], \

```

```

77     double omega[N],double k[N],double A[N],    \
78     double H3[N][NL],double H5[N][NL],        \
79     double delta3[N][NL],double delta5[N][NL],\
80     double epsilon[N]);
81
82 void kontroll();

```

Listing 18: dmet.sh

```

1  #Kompilerer hovedprogram
2  make clean
3  make
4  ./fprogram
5
6  make c
7  ./cprogram
8
9  echo -----
10 echo Hei og velkommen til DMET DELUXE 2000
11 echo Vennligst følg instruksene på skjermen
12 echo Trykk j for ja, eller ENTER for nei
13 echo -----
14
15 read -p "Vil du plotte resultatgrafene fra programmet? (j/n) " answer
16 case ${answer:0:1} in
17     j|J )
18         #Starter gnuplot
19         gnuplot -persist << __EOF
20 set yrange[0:50]
21 set term wxt 7
22 set title "Beregnet JONSWAP-spekter"
23 set ylabel "Sn"
24 plot "spekterplot.dat" with lines title "JONSWAP"
25
26 set yrange [0:4]
27 set xrange [0:2]
28 set term wxt 1
29 set title "Tilleggsmasse for begge lektere"
30 set xlabel "omega (B/2g)^0.5 [-]"
31 set ylabel "A33 2D"
32 plot "A33plot.txt" with lines title "A33 lekter A"
33 replot "A33plot.txt" using 3:4 with lines title "A33 lekter B"
34
35 set yrange [-0.2:1.1]
36 set term wxt 2
37 set xzeroaxis linetype 3 linewidth 2
38 set title "Dempekoeffisient for begge lektere"
39 set ylabel "B33 2D"
40 plot "B33plot.txt" with lines title "B33 lekter A"
41 replot "B33plot.txt" using 3:4 with lines title "B33 lekter B"
42 unset xzeroaxis
43
44 set yrange[0:1.1]
45 set xrange[0:3]
46 set term wxt 20
47 set title "Transferfunksjon hiv for begge lektere"

```

```

48 set xlabel "omega [rad/s]"
49 set ylabel "|H3(omega)| [-]"
50 plot "H3plot.txt" with lines title "H3 leker A [-]"
51 replot "H3plot.txt" using 1:3 with lines title "H3 leker B [-]"
52
53 set yrange[0:1.1]
54 set term wxt 21
55 set ylabel "|H5(omega)| [-]"
56 set title "Transferfunksjon stamp for begge leker"
57 plot "H5plot22.txt" with lines title "H5 leker A [-]"
58 replot "H5plot22.txt" using 1:3 with lines title "H5 leker B [-]"
59
60 set yrange[-4:0]
61 set term wxt 11
62 set xzeroaxis
63 set ylabel "delta3 [-]"
64 set title "Fasevinkel hiv for leker A"
65 plot "fase3sjekk.txt" with lines title "Delta3 for leker "
66 replot "fase3sjekk.txt" using 1:3 with lines title "Delta3 for leker B"
67
68 set term wxt 12
69 set yrange[0:4]
70 set ylabel "delta5 [-]"
71 set title "Fasevinkel stamp for leker A"
72 plot "fase5sjekk.txt" with lines title "Delta5 for leker A"
73 replot "fase5sjekk.txt" using 1:3 with lines title "Delta5 for leker B"
74 unset xzeroaxis
75
76 __EOF
77 ;;
78 * )
79 echo OK, fortsetter kjøring
80 ;;
81 esac
82
83 read -p "Vil du ha en utskrift fra bølgestatistikken i programmet?
      (j/n) " answer
84 case ${answer:0:1} in
85     j|J )
86         #skriver til skjerm
87         cat statistikk.txt << __EOF
88
89 __EOF
90 ;;
91 * )
92 echo OK, kjører visualiseringsprogrammet
93 ;;
94 esac
95
96 read -p "Vil du plotte respons og bølge for regulær sjø? (j/n) " answer
97 case ${answer:0:1} in
98     j|J )
99         #plotter
100         gnuplot -persist << __EOF
101 set term wxt 15
102 set title "Respons mot overflateheving, leker A, omega = 0.399"
103 set ylabel "Utslag [m]"

```

```
104 set xlabel "Tid [s]"
105 plot "overflateplott.dat" using 1 with lines title "zeta", \
106      "responsplott.dat" every ::1::600 using 1 with lines title "eta3", \
107      "responsplott.dat" every ::1::600 using 2 with lines title "eta5"
108 __EOF
109 ;;
110 * )
111 echo Programmet er ferdig, takk for i dag!
112 ;;
113 esac
```