Hovedfagsoppgave i hydroakustikk

Design og studier av lydkilde for SOBER-prosjektet

Audun Pedersen

Fysisk institutt
Universitetet i Bergen
mai 2002

Forord

Dette hovedfagsarbeidet i hydroakustikk ble i sin helhet utført på Fysisk institutt ved Universitetet i Bergen. Oppgaven ble påbegynt vårsemesteret 2001, og det har dermed tatt tre semester å fullføre den.

Arbeidet har i grove trekk bestått av tre deler. Det første semesteret gikk med til hovedfagspensum og innføring i Bergenkoden. Deretter ble lydkilden konstruert og laget. I denne fasen gjorde overingeniør Kåre Slettebakken ved instituttets mekaniske verksted et viktig arbeid. Lydfeltet fra den ferdige transduseren ble til slutt undersøkt og vurdert i forhold til lineær og ikkelineær teori.

Jeg vil takke professor Halvor Hobæk for god veiledning, og for at jeg fikk anledning til å delta på de årlige skandinaviske symposiene i fysikalsk akustikk (SSPA) på Ustaoset i 2001 og 2002. Den siste gangen presenterte jeg deler av innholdet i denne oppgaven.

Førsteamanuensis Magne Vestrheim har bidratt med gode råd og tips, og medstudentene mine har hjulpet meg med sine erfaringer i tillegg til å skape et godt sosialt miljø på skolen.

Jeg vil også takke familie og venner, og spesielt mine foreldre Svein Pedersen og Turid Elise Kindt Pedersen, for deres hjelp og moralske støtte gjennom hele studietiden.

Bergen, 15. mai 2002

Audun Pedersen

Innledning

Når en lydkilde beveger seg langs en rett linje i et homogent medium, med fart større enn lydfarten i mediet, dannes en kjegleformet sjokkfront med toppunkt i kilden.

Sjokket er en N-bølge. Den består av et innledende sjokk som øker trykket fra likevektstrykket til et positivt overtrykk, en lineær dekompresjon og et halesjokk som øker trykket tilbake til likevekt [1].

Når lydkilden akselererer eller endrer kurs, vil det kunne observeres et fokusert overlydssmell som kjennetegnes ved en U-formet bølge [2]. I en U-bølge leder både front- og halesjokket til positive overtrykk. Det maksimale overtrykket er typisk to til fem ganger større enn i en N-bølge fra samme kilde.

Energien i sjokkbølgen fra et overlydsfly er fordelt over frekvenser fra 0,1 Hz til 100 Hz. Smellet som observeres på bakken varer i 100–500 ms, avhengig av flyets størrelse.

For at sjokkbølger ikke skal nå befolkede områder, er flygning i overlydsfart forbudt over de fleste landområder. Hvis forbudssonenes grenser blir trukket for langt fra kysten, kan regelverket true lønnsomheten til kommersiell overlydstransport [3].

SOBER (Sonic Boom European Research Programme) er en del av EUs femte rammeprogram. Prosjektet har som mål å utarbeide regnemodeller som gir et best mulig grunnlag for å legge flyruter slik at overlydsflygninger ikke skaper problemer i kystområder eller rundt øyer. Det blir lagt spesiell vekt på fokuserte sjokk, skyggesonen som oppstår på grunn av atmosfærens stratifisering, og såkalte sekundære sjokk.

I Bergen skal det gjøres et laboratorie-eksperiment der man studerer forplantningen av akustiske sjokkbølger i en stratifisert blanding av vann og etanol. Forsøkstanken vil fungere som en modell av atmosfæren opp til ca. 15 kilometers høyde.

I denne oppgaven ble det konstruert en lydkilde til eksperimentet. Det ble også gjort målinger i transduserens utstrålte lydfelt, og resultatet ble sammenlignet med teoretiske verdier.

Innhold

Kapittel 1

Teori

1.1 Piezoelektriske elementer

Et piezoelektrisk materiale kjennetegnes ved den direkte og den inverse piezoelektriske effekten. Når det utsettes for mekaniske spenninger i bestemte retninger, oppstår en elektrisk polarisering i materialet. Omvendt, hvis et elektrisk felt blir satt opp i bestemte retninger i stoffet, vil dette forårsake en mekanisk deformasjon (den inverse piezoelektriske effekten). Sammenhengen mellom deformasjon og elektrisk polarisasjon blir her betraktet som lineær.

I denne oppgaven ble det benyttet skiver av bly-zirkonat-titanat-materialet Pz26 fra Ferroperm. Elementene er polarisert over Curie-temperaturen ved hjelp av elektriske felt på 1–4 kV/mm. Teorien som er tatt med her, er i hovedsak hentet fra forelesningsnotater skrevet av førsteamanuensis Magne Vestrheim for emnet FYS 272 [4].

1.1.1 Mekaniske egenskaper

Tøyning

De mekaniske egenskapene til faste stoffer kan beskrives lineært ved hjelp av tøynings- og spenningstensorer. Tøyningstensoren $\tilde{\mathbf{S}}$ gir et mål på deformasjonen i et stoff,

$$\tilde{\mathbf{S}} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} = \frac{1}{2}(\nabla\boldsymbol{\xi} + (\nabla\boldsymbol{\xi})^{\mathrm{T}}). \tag{1.1}$$

$\boldsymbol{\xi} = \boldsymbol{\xi}(\mathbf{x}, t)$ er forskyvningen ved tiden t av den delen av materialet som befinner seg i punktet $\mathbf{x} = (x, y, z)$ ved likevekt (når $S_{ij} = 0$, $i, j = 1, 2, 3$).

Som ligning (1.1) viser, er tøyningstensoren den symmetriske delen av gradienten til forskyvningen. Den antisymmetriske delen $\frac{1}{2}(\nabla\boldsymbol{\xi} - (\nabla\boldsymbol{\xi})^{\mathrm{T}})$ svarer til lokal rotasjon og blir her neglisjert.

Tøyningskomponentene på diagonalen,

$$S_{ij} = \frac{\partial \xi_i}{\partial x_j}, \quad i = 1, 2, 3, \quad j = i,$$

er normaltøyningskomponenter, mens resten av komponentene kalles skjærtøyningskomponenter,

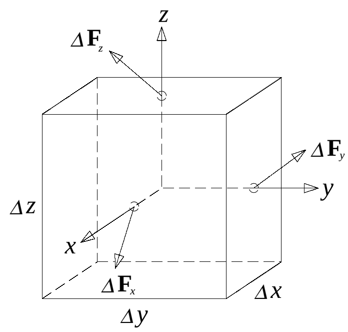
$$S_{ij} = \frac{1}{2}\left(\frac{\partial \xi_i}{\partial x_j} + \frac{\partial \xi_j}{\partial x_i}\right), \quad i, j = 1, 2, 3, \quad i \neq j.$$

Siden tøyningstensoren er symmetrisk, kan det innføres en forkortet vektornotasjon,

$$\mathbf{S} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{bmatrix} = \begin{bmatrix} S_{11} \\ S_{22} \\ S_{33} \\ 2S_{23} \\ 2S_{13} \\ 2S_{12} \end{bmatrix}.$$

Spenning

Kreftene som virker på et volumelement som på figur 1.1 kan beskrives ved hjelp av spenningstensoren $\tilde{\mathbf{T}}$. På flaten normalt på x-aksen virker kraften $\Delta\mathbf{F_x} = \mathbf{T}_1 \Delta A_1$, der $\Delta A_1 = \Delta y\, \Delta z$. Spenningsvektoren har tre komponenter, $\mathbf{T}_1 = \mathbf{i}T_{11} +$



Figur 1.1: De ytre kreftene som virker på et kubisk volumelement.

$\mathbf{j}T_{21} + \mathbf{k}T_{31}$. De tre spenningsvektorene $\mathbf{T}_1$, $\mathbf{T}_2$ og $\mathbf{T}_3$ utgjør til sammen spenningstensoren,

$$\tilde{\mathbf{T}} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix}.$$

Når det antas at spenningstensoren er symmetrisk, kan den skrives som en vektor på lignende måte til $\tilde{\mathbf{S}}$,

$$\mathbf{T} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix} = \begin{bmatrix} T_{11} \\ T_{22} \\ T_{33} \\ T_{23} \\ T_{13} \\ T_{12} \end{bmatrix}. \tag{1.2}$$

Generalisert Hookes lov

Ved små forskyvninger er sammenhengen mellom tøyning og spenning lineær, dvs.

$$T_{ij} = c_{ijkl} S_{kl}, \quad S_{ij} = s_{ijkl} T_{kl}, \quad i,j,k,l = 1,2,3. \tag{1.3}$$

Dette er en generalisert form av Hookes lov. For stivhetskonstantene c_{ijkl} gjelder det at

$$c_{ijkl} = c_{jikl} = c_{ijlk} = c_{jilk}, \tag{1.4}$$

og tilsvarende for komplianskonstantene s_{ijkl}. (1.3) kan da skrives i den forkortede notasjonen,

$$T_p = c_{pq} S_q, \quad S_p = s_{pq} T_q, \quad p,q = 1,2,\ldots,6. \tag{1.5}$$

Hver p- eller q-verdi tilsvarer et tallpar (i,j) eller (k,l), henholdsvis, på samme måte som i ligning (1.2). Videre er det mulig å vise at $c_{pq} = c_{qp}$ og $s_{pq} = s_{qp}$.

For et piezokeramisk materiale som er polarisert i z-retningen, er koeffisientmatrisene ved konstant elektrisk felt

$$[c^E] = \begin{bmatrix} c_{11}^E & c_{12}^E & c_{13}^E & 0 & 0 & 0 \\ c_{12}^E & c_{22}^E & c_{13}^E & 0 & 0 & 0 \\ c_{13}^E & c_{13}^E & c_{33}^E & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44}^E & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{44}^E & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}(c_{11}^E - c_{12}^E) \end{bmatrix}$$

og

$$[s^E] = \begin{bmatrix} s_{11}^E & s_{12}^E & s_{13}^E & 0 & 0 & 0 \\ s_{12}^E & s_{22}^E & s_{13}^E & 0 & 0 & 0 \\ s_{13}^E & s_{13}^E & s_{33}^E & 0 & 0 & 0 \\ 0 & 0 & 0 & s_{44}^E & 0 & 0 \\ 0 & 0 & 0 & 0 & s_{44}^E & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(s_{11}^E - s_{12}^E) \end{bmatrix}.$$

1.1.2 Elektriske egenskaper

De piezoelektriske skivene som ble brukt i dette arbeidet, har tynne elektroder på endeflatene, dvs. vinkelrett på z-aksen. Når det antas at arealet er stort i forhold til tykkelsen på skiven, gir en potensialforskjell mellom elektrodene et tilnærmet homogent elektrisk felt i z-retningen. Da lydfartene i materialet er mye lavere enn forplantningsfarten for elektromagnetiske bølger, kan man bruke den elektrostatiske relasjonen mellom elektrisk forskyvning, elektrisk felt og polarisasjon (kvasi-elektrostatisk approksimasjon [4]),

$$\mathbf{D} = \varepsilon_0 \mathbf{E} + \mathbf{P} = \tilde{\varepsilon}\mathbf{E}.$$

ε_0 og $\tilde{\varepsilon}$ er henholdsvis permittiviteten i vakuum og permittivitetstensoren i det piezoelektriske mediet. Ved konstant tøyning kan permittivitetstensoren for et piezokeramisk materiale polarisert i z-retningen skrives

$$[\varepsilon^S] = \begin{bmatrix} \varepsilon_{11}^S & 0 & 0 \\ 0 & \varepsilon_{11}^S & 0 \\ 0 & 0 & \varepsilon_{33}^S \end{bmatrix}.$$

I et statisk felt er også

$$\nabla \cdot \mathbf{D} = 0,$$

og hvis det ikke finnes frie ladninger i mediet, er

$$\nabla \times \mathbf{E} = \mathbf{0}.$$

1.1.3 Piezoelektriske relasjoner

Den piezoelektriske koblingen i et materiale kan beskrives ved hjelp av et sett av lineære konstitutive ligninger. Relasjonene kan skrives på flere måter ([4],[5]), for eksempel

$$\begin{aligned} T_p &= c_{pq}^E S_q - e_{jp} E_j, \\ D_i &= e_{iq} S_q - \varepsilon_{ij}^S E_j, \quad p, q = 1, 2, \ldots, 6, \; i, j = 1, 2, 3. \end{aligned} \tag{1.6}$$

Disse ligningene er skrevet etter Einsteins sumkonvensjon, slik at det skal summeres over alle indekser som forekommer to ganger i et ledd.

e_{ip} er de piezoelektriske spenningskonstantene. For en keramisk skive polarisert i z-retningen, kan de skrives

$$[e] = \begin{bmatrix} 0 & 0 & 0 & 0 & e_{15} & 0 \\ 0 & 0 & 0 & e_{15} & 0 & 0 \\ e_{31} & e_{31} & e_{33} & 0 & 0 & 0 \end{bmatrix}.$$

Spenningskonstantene er forhold mellom endring i mekanisk spenning og elektrisk felt ved konstant tøyning, og samtidig mellom endring i elektrisk forskyvning og mekanisk tøyning ved konstant elektrisk felt. Enheten for e_{ip} er C/m^2.

Til sammen behøves 10 uavhengige parametere for å beskrive egenskapene til et keramisk piezoelektrisk materiale, for eksempel:

$$c_{11}^E, c_{12}^E, c_{13}^E, c_{33}^E, c_{44}^E, e_{15}, e_{31}, e_{33}, \varepsilon_{11}^S \text{ og } \varepsilon_{33}^S. \tag{1.7}$$

1.1.4 Tykkelsessvingninger i en piezoelektrisk skive

Bølgeligning

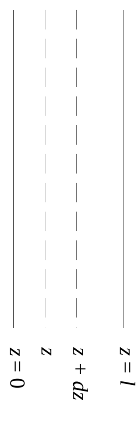
En forenklet beskrivelse av tykkelsessvingningene i en skive av piezoelektrisk keramikk kan finnes når forholdet mellom skivens diameter og tykkelse er stort, $d/\ell \gg 1$.

Man antar at all bevegelse i skiven er i z-retningen og at alle størrelser er konstante med hensyn på x og y,

$$\boldsymbol{\xi} = \xi_3 \mathbf{k}, \quad \frac{\partial}{\partial x} = \frac{\partial}{\partial y} = 0. \tag{1.8}$$

Tøyningen $\mathbf{S}$ blir da gitt ved

$$S_3 = \frac{\partial \xi_3}{\partial z}, \quad S_1 = S_2 = S_4 = S_5 = S_6 = 0. \tag{1.9}$$



Figur 1.2: Skiveelement med tykkelse dz.

På et tynt lag med posisjon z og tykkelse dz (figur 1.2), er nettokraften

$$\Delta F = A(T_3(z + dz) - T_3(z)) = A \frac{\partial T_3}{\partial z} dz.$$

Skiveelementet har masse $dm = \rho_m A\,dz$. Innsetting i Newtons 2. lov gir

$$A\frac{\partial T_3}{\partial z}dz = \rho_m A\,dz\frac{\partial^2 \xi}{\partial t^2}, \tag{1.10}$$

der forskyvningen ξ er i z-retningen. Av (1.6) følger det at

$$T_3 = c_{33}^D S_3 - \frac{e_{33}}{\varepsilon_{33}^S}D_3, \quad c_{33}^D = c_{33}^E(1 + \frac{e_{33}^2}{\varepsilon_{33}^S c_{33}^E}). \tag{1.11}$$

Ved å dividere ligning (1.10) med $A\,dz$ og sette inn for $S_3 = \partial\xi/\partial z$, og T_3 når $\partial D/\partial z = 0$, oppnås den endimensjonale bølgeligningen

$$\frac{\partial^2 \xi}{\partial z^2} - \frac{1}{c_m^2}\frac{\partial^2 \xi}{\partial t^2} = 0. \tag{1.12}$$

$c_m = \sqrt{c_{33}^D/\rho_m}$ er lydfarten for longitudinale bølger i z-retningen.

Den generelle løsningen av ligning (1.12) kan skrives som en superposisjon av to enkle plane bølger i henholdsvis positiv og negativ retning,

$$\xi(z,t) = Ge^{i(\omega t - k_m z)} + He^{i(\omega t + k_m z)}. \tag{1.13}$$

Bølgetallet k_m er gitt ved

$$k_m = \frac{\omega}{c_m}.$$

Randbetingelser

Når partikkelfarten $u = \partial\xi/\partial t$ skrives på formen $u = u_0 e^{i\omega t}$, blir grensebetingelsene på skivens endeflater

$$z = 0: \quad u_1 = i\omega(G + H)e^{i\omega t}, \tag{1.14}$$

$$z = \ell: \quad u_2 = i\omega(Ge^{-ik_m\ell} + He^{ik_m\ell})e^{i\omega t}. \tag{1.15}$$

Ved innsetting i (1.15) av H og G fra (1.14) etter tur, fås henholdsvis

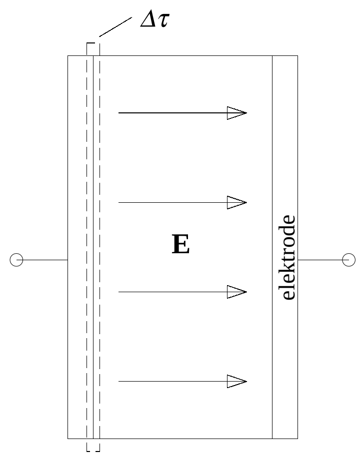
$$G = \frac{u_{2,0} - u_{1,0}e^{ik_m\ell}}{2\omega \sin k_m\ell}, \quad H = \frac{u_{1,0}e^{-ik_m\ell} - u_{2,0}}{2\omega \sin k_m\ell}. \tag{1.16}$$

Summen av krefter som virker på en endeflate med infinitesimal tykkelse dz, er lik null, slik at grensebetingelsene for krefter på endeflatene blir

$$F_1 = -(T_3)_{z=0}A, \quad F_2 = (T_3)_{z=\ell}A, \tag{1.17}$$

der de ytre kreftene $\mathbf{F}_1$ og $\mathbf{F}_2$ begge peker mot høyre på figur 1.2. Ligning (1.11) og den generelle løsningen (1.13) med $\xi = \xi_0 e^{i\omega t}$ og $T = T_0 e^{i\omega t}$ gir

$$T_0 A = -Z_{0m} i\omega(Ge^{-ik_m z} - He^{ik_m z}), \tag{1.18}$$



Figur 1.3: Integrasjonsvolum for den elektriske randbetingelsen.

der $Z_{0m} = \rho_m c_m A$. Innsetting av G og H fra ligning (1.16) og definisjon av de fire størrelsene

$$C_0 = \frac{\varepsilon_{33}^S A}{\ell}, \quad \phi = \frac{e_{33} A}{\ell}, \tag{1.19}$$

$$Z_a = i Z_{0m} \tan \frac{k_m \ell}{2} \quad \text{og} \quad Z_b = \frac{Z_{0m}}{i \sin k_m \ell}, \tag{1.20}$$

gir følgende uttrykk for kreftene på skivens endeflater:

$$\begin{aligned} F_1 &= Z_a u_1 + Z_b(u_1 - u_2) + \frac{\phi}{i\omega C_0} I, \\ -F_2 &= -Z_a u_2 + Z_b(u_1 - u_2) + \frac{\phi}{i\omega C_0} I. \end{aligned} \tag{1.21}$$

Den utvidede Ampères lov kan skrives

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}_\mathrm{C}, \tag{1.22}$$

der $\mathbf{J}_\mathrm{C}$ er strømtettheten for ledningsstrøm. Ved å ta divergensen av (1.22) og integrere over et lite volum $\Delta\tau$ som omslutter grenseflaten i $z = 0$ mellom elektroden og den piezoelektriske skiven (figur 1.3), finnes

$$I = -\int_A \mathbf{J}_\mathrm{C} \cdot d\mathbf{A} = \frac{\partial D_3}{\partial t} A.$$

Når $D_3 = D_0 e^{i\omega t}$, kan uttrykket ovenfor skrives

$$I = i\omega D_3 A. \tag{1.23}$$

Det piezoelektriske materialet kan ses på som et rent dielektrikum, slik at strømtettheten $\mathbf{J}_C$ inne i skiven er null. I må da være strømmen som føres inn på elektroden gjennom en elektrisk tilkobling.

Spenningen over transduserelementet kan finnes fra ligning (1.23 og den andre av ligningene (1.6):

$$V = \int_0^\ell E_3\,dz = \int_0^\ell \left(\frac{D_3}{\varepsilon_{33}^S} - \frac{e_{33}}{\varepsilon_{33}^S}\frac{\partial \xi_3}{\partial z}\right) dz.$$

Ved å kombinere dette uttrykket med ligning (1.19) og (1.23), fås

$$V = \frac{1}{i\omega C_0} I - \frac{\phi}{i\omega C_0}(u_2 - u_1). \tag{1.24}$$

Ligningene (1.21) og (1.24) danner til sammen et sekspolsystem med symmetrisk koeffisientmatrise,

$$\begin{bmatrix} F_1 \\ F_2 \\ V \end{bmatrix} = \begin{bmatrix} (Z_a + Z_b) & -Z_b & \frac{\phi}{i\omega C_0} \\ -Z_b & (Z_a + Z_b) & -\frac{\phi}{i\omega C_0} \\ \frac{\phi}{i\omega C_0} & -\frac{\phi}{i\omega C_0} & \frac{1}{i\omega C_0} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ I \end{bmatrix}. \tag{1.25}$$

Fra (1.19) kan C_0 gjenkjennes som kapasitansen skiven har når de mekaniske svingningene er ideelt blokkert. Faktoren ϕ knytter strømstyrken til partikkelhastigheten på endeflatene og beskriver på denne måten den piezoelektriske koblingen i skiven. Fra de tre ligningene kan man se at [4]

$$\phi u = I, \quad Z_{EL} = \frac{Z_{mek}}{\phi^2},$$

der Z_{EL} er den ekvivalente elektriske impedansen til den mekaniske impedansen Z_{mek}. I resten av dette arbeidet vil impedanser der indeksen er skrevet med små bokstaver være mekaniske, mens store bokstaver betegner de elektriske ekvivalente impedansene.

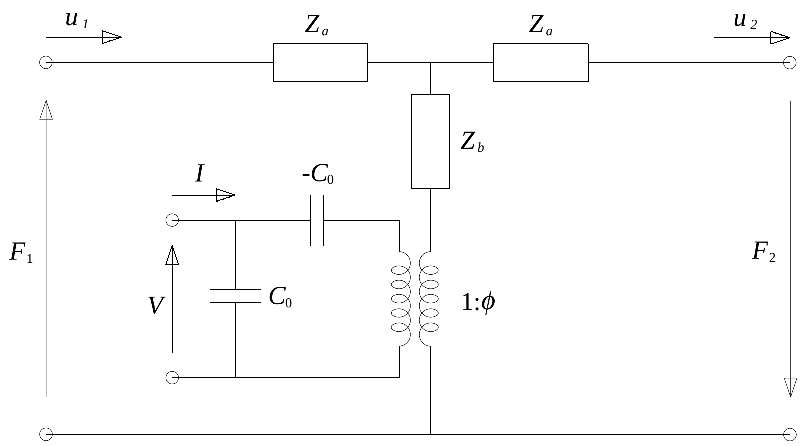
Mason-modellen, frie svingninger i vakuum

Ligningssettet (1.25) kan representeres ved ekvivalentkretsen i figur 1.4. Den piezoelektriske koblingen tilsvarer her en ideell transformator med transformasjonsforhold $1 : \phi$. Kretsen har fire mekaniske og to elektriske poler.

Når elementet svinger fritt, kan kretsen reduseres til den på figur 1.5. I figur 1.5b er de mekaniske impedansene erstattet av elektriske ekvivalente impedanser,

$$Z_A = \frac{Z_a}{\phi^2}, \quad Z_B = \frac{Z_b}{\phi^2}.$$

Kretsens elektriske admittans $Y_T = 1/Z_T$ er i dette tilfellet gitt ved



Figur 1.4: Mason ekvivalentskjema for piezokeramisk skive [4].

$$Y_T = i\omega C_0 + \frac{1}{\frac{i}{\omega C_0} + Z_B + \frac{Z_A}{2}}. \tag{1.26}$$

Ved hjelp av definisjonene av Z_a og Z_b (1.20) og den matematiske relasjonen

$$\tan\frac{\theta}{2} = \frac{1}{\sin\theta} - \frac{1}{\tan\theta} = \frac{\sin\theta}{1+\cos\theta}$$

kan ligning (1.26) omformes til

$$Y_T = i\frac{\omega C_0}{1 - k_t^2 \frac{\tan(\frac{1}{2}k_m\ell)}{\frac{1}{2}k_m\ell}} = iB_T, \quad G_T = 0. \tag{1.27}$$

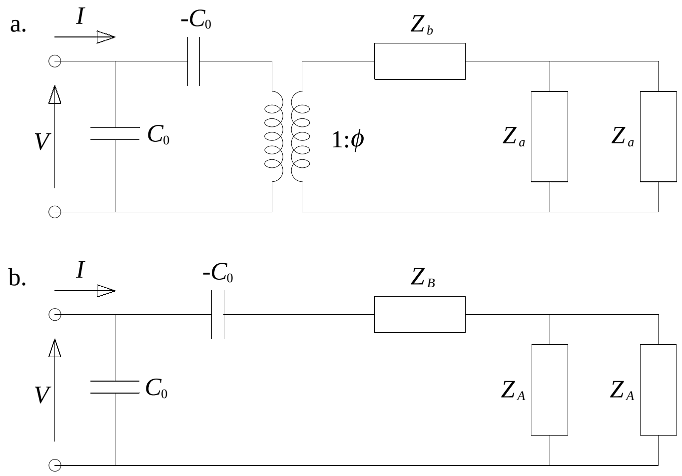
$k_t = k_{33}^t = \frac{e_{33}}{\sqrt{\varepsilon_{33}^S c_{33}^D}}$ er koblingsfaktoren for rene tykkelsessvingninger [5]. Admittansen er kompleks og kan skrives på formen $Y_T = G_T + iB_T$, der G_T er konduktansen og B_T susceptansen i kretsen.

Impedansen $Z_T = R_T + iX_T$ som tilsvarer admittansen i (1.27), kan skrives

$$Z_T = \frac{1}{i\omega C_0}(1 - k_t^2 \frac{\tan(\frac{1}{2}k_m\ell)}{\frac{1}{2}k_m\ell}) = iX_T, \quad R_T = 0. \tag{1.28}$$

1.1.5 Karakteristiske frekvenser

Hittil er det blitt sett bort fra tap i det piezoelektriske materialet. I IEEE-standard 176-1987 [5] er resonansfrekvensen f_1 og antiresonans-frekvensen f_2 i



Figur 1.5: Mason ekvivalentskjema for en piezokeramisk skive i vakuum, a) med mekaniske impedanser, b) med elektriske ekvivalente komponenter.

det tapsfrie tilfellet definert ved

$$\begin{aligned} Z_T(f = f_1) &= 0, \quad Y_T(f = f_1) = \infty, \\ Z_T(f = f_2) &= \infty, \quad Y_T = 0. \end{aligned}$$

Som ligning (1.28) viser, blir $Z_T = 0$ når

$$k_t^2 \frac{\tan(\frac{1}{2}k_m\ell)}{\frac{1}{2}k_m\ell} = 1.$$

Dette betyr at overtonene ikke blir harmoniske (resonans inntreffer ikke ved hele antall grunnfrekvenser). Overtoner som er høye nok blir tilnærmet harmoniske, og det samme skjer for små k_t.

Frekvensene som gir antiresonans tilfredsstiller

$$\frac{1}{2}k_m\ell = \frac{\pi}{2}. \tag{1.29}$$

Bølgelengden $\lambda = c_m/f = 2\pi/k_m$, slik at ligning (1.29) blir tilfredsstilt når $\lambda = 2\ell$.

Når man tar hensyn til tap i elementet, får impedansen en endelig realdel. De karakteristiske frekvensene blir da forbundet med hver sine tre frekvenser (tabell 1.1). IEEE-standarden anbefaler at f_s og f_p brukes for henholdsvis resonans- og antiresonansfrekvensen når den tapsfrie teorien brukes på et reelt problem.

1.2 Mekanisk og elektrisk impedanstilpasning

På grunn av dimensjonene på SOBER-prosjektets forsøkstank, var det ønskelig at den konstruerte transduseren kunne sende ut korte, velformede lydpulser, dvs. ha en kort transientrespons. Dette oppnås ved å øke kildens båndbredde. Ifølge DeSilets, Fraser og Kino [6] gir en overføringsfunksjon med lineær fase og gaussisk formet absoluttverdi den korteste impulsresponsen.

1.2.1 Elektrisk tilkobling

Signalkabel

Den karakteristiske impedansen i en kabel er definert som $Z_{0\mathrm{c}} = \sqrt{L_\mathrm{c}/C_\mathrm{c}}$, der L_c og C_c er kabelens induktans og kapasitans per lengdeenhet. Impedansen Z_INN inn på kabelen er gitt ved kabelens karakteristiske impedans, termineringsimpedansen Z_L og kabelens lengde ℓ_c,

$$Z_\mathrm{INN} = \frac{Z_\mathrm{L} + iZ_{0\mathrm{c}} \tan(k_\mathrm{EM}\ell_\mathrm{c})}{Z_{0\mathrm{c}} + iZ_\mathrm{L} \tan(k_\mathrm{EM}\ell_\mathrm{c})} Z_{0\mathrm{c}}. \tag{1.30}$$

$k_\mathrm{EM} = \omega/c_\mathrm{EM} = \omega\sqrt{L_\mathrm{c}C_\mathrm{c}}$ er bølgetallet for elektromagnetiske bølger i kabelen.

Hvis $k_\mathrm{EM}\ell_\mathrm{c}$ er liten nok, er $\tan(k_\mathrm{EM}\ell_\mathrm{c}) \approx k_\mathrm{EM}\ell_\mathrm{c}$, og (1.30) reduseres til

$$Z_\mathrm{INN} = \frac{1 + i\frac{Z_{0\mathrm{c}}}{Z_\mathrm{L} k_\mathrm{EM}\ell_\mathrm{c}}}{1 + i\frac{Z_\mathrm{L}}{Z_{0\mathrm{c}}} k_\mathrm{EM}\ell_\mathrm{c}} Z_\mathrm{L}.$$

For lave frekvenser og korte kabler kan man se bort fra kabelens induktans, og uttrykket for kabelens inngangsimpedans blir

$$Z_\mathrm{INN} \approx \frac{Z_\mathrm{L}}{1 + iZ_\mathrm{L}\omega C_\mathrm{c}\ell_\mathrm{c}}. \tag{1.31}$$

Tabell 1.1: Resonansfrekvenser for tykkelsesmoden i et piezoelektrisk element. Anbefalte frekvenser for substitusjon i tapsfri teori [4],[5].

Uten tap		Med tap		Erstatning
Resonans	f_1	Maksimal admittans, $d\lvert Y_T\rvert/d\omega = 0$	f_m	f_s
		Serieresonans, $X_M = 0$		
		IEEE 176-1987: Maksimal konduktans	f_s	
		Resonans, $X_T = B_T = 0$	f_r	
Antiresonans	f_2	Antiresonans($f_a > f_r$), $X_T = B_T = 0$	f_a	f_p
		Parallellresonans, $X_T\rvert_{R_M=0} = \infty$		
		IEEE 176-1987: Maksimal resistans	f_p	
		Minimal admittans, $d\lvert Y_T\rvert/d\omega = 0$	f_n	

Som uttrykket viser, kan kabelen under forutsetningene som er gjort ovenfor ses på som en kapasitans $C_c \ell_c$ i parallell med lasteimpedansen.

Elektrisk tilpasningsenhet

Den maksimale effektoverføringen oppnås når kretsen som skal drives av effektforsterkeren har impedans lik komplekskonjugatet av forsterkerens utgangsimpedans. Denne er på 50 Ω og tilnærmet reell.

Piezoelektriske elementer har som regel kapasitiv reaktans i frekvensområdet rundt parallellresonansen. Den reaktive delen av transduserens impedans fjernes derfor ved å sette en induktans inn i kretsen, enten i serie eller i parallell med elementet. Det er også vanlig å justere realdelen av impedansen ved hjelp av en transformatorkobling.

Et alternativ til transformatorløsningen er $\pi-$filteret [7], der både den resistive og den reaktive delen av impedansen justeres ved hjelp av to kondensatorer og en spole (figur 1.6). Inngangsimpedansen $Z_{\text{INN}} = R_{\text{INN}} + iX_{\text{INN}}$ er gitt ved

$$\begin{aligned}
R_{\text{INN}} &= \frac{R_{\text{UT}}}{[(1-\omega^2 LC_1) - \omega C X_{\text{UT}}]^2 + (R_{\text{UT}}\omega C)^2}, \\
X_{\text{INN}} &= \frac{\omega^2(1-\omega^2 LC_1) + X_{\text{UT}}(1-2\omega^2 LC) - \omega C(1-\omega^2 LC_2)(|Z_{\text{UT}}|^2}{[(1-\omega^2 LC_1) - \omega C X_{\text{UT}}]^2 + (R\omega C)^2},
\end{aligned} \tag{1.32}$$

der $C = C_1 + C_2 - \omega^2 LC_1C_2$ og $Z_{\text{UT}} = R_{\text{UT}} + iX_{\text{UT}}$.

Hvis L og C_1 velges slik at $\omega^2 LC_1 = 1$, er $C = C_1$ og inngangsimpedansen kan skrives

$$R_{\text{INN}} = \frac{R_{\text{UT}}}{\omega^2 C_1^2 |Z_{\text{UT}}|^2}, \quad X_{\text{INN}} = \frac{-X_{\text{UT}} - \omega(C_1 - C_2)|Z_{\text{UT}}|^2}{\omega^2 C_1^2 |Z_{\text{UT}}|^2}. \tag{1.33}$$

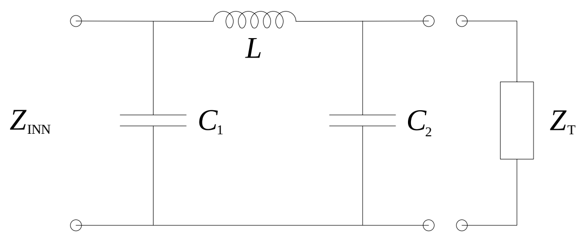
Som (1.33) viser, kan den reelle delen av impedansen justeres ved å velge C_1 og L. Deretter kan C_2 settes slik at $X_{\text{INN}} = 0$ ved den ønskede frekvensen.

1.2.2 Mekanisk koblingslag

Piezokeramiske elementer som sender lyd direkte ut i vann har relativt smal båndbredde og dermed en langvarig impulsrespons. Dette skyldes blant annet den store forskjellen i spesifikk akustisk impedans mellom vann (ca. 1,5 MRayl) og keramiske materialer (25–35 MRayl).

Ved å plassere et koblingslag foran kilden, oppnås en bedret effektoverføring og dermed en større båndbredde enn den elementet har alene [6],[8],[9]. Impedanstilpasningen kan bedres videre ved å ta i bruk to eller flere koblingslag [10],[11].

Båndbredden til en transduser kan også økes ved å plassere ett eller flere media med lav kvalitetsfaktor bak den piezoelektriske skiven [6],[12],[13]. En ulempe med denne fremgangsmåten, er at mye effekt går tapt i baklaget.



Figur 1.6: $\pi-$filter.

Transmisjonslinjemodell

For en tynn elastisk skive med areal A, tykkelse ℓ og bevegelse kun i tykkelsesretningen, gjelder den samme bølgeligningen som for det piezoelektriske elementet (1.12),

$$\frac{\partial^2 \xi}{\partial z^2} - \frac{1}{c_k^2}\frac{\partial^2 \xi}{\partial t^2} = 0.$$

Istedenfor ligning (1.11) brukes i dette tilfellet den generaliserte Hookes lov (1.5), slik at

$$c_k = \sqrt{\frac{c_{33}}{\rho_k}}.$$

Ved å sette $\xi = \xi_0 e^{i\omega t}$, $T = T_0 e^{i\omega t}$ og Hookes lov inn i den generelle løsningen (1.13), finnes

$$T_0 A = -i\omega\, Z_{0k}(G e^{-ik_k z} - H e^{ik_k z}), \tag{1.34}$$

der $Z_{0k} = \rho_k c_k A$.

Det er også mulig å ta med tap i modellen. Impedansen kan da skrives [14]

$$Z_{0k} = A\left(\frac{\rho_k c_k}{1+r^2} + i\frac{r\rho_k c_k}{1+r^2}\right),$$

der tapsfaktoren $r = \alpha c_k/\omega$, og α er absorpsjonskoeffisienten i Np/m. En annen måte å inkludere tap på, er å gjøre c_{33}^D og ε_{33}^S komplekse [15],

$$\hat{c}_{33}^D = c_{33}^D(1 + \frac{i}{Q_m}), \quad \hat{\varepsilon}_{33}^S = \varepsilon_{33}^S(1 - i\tan\delta_e).$$

Her er Q_m den mekaniske kvalitetsfaktoren og δ_e tapsvinkelen for dielektrisk tap.

På endeflatene gjelder grensebetingelsene

$$F_2 = (T_3)_{z=\ell}A, \quad F_1 = (T_3)_{z=0}A.$$

Ved å sette grenseverdiene inn i (1.34), oppnås

$$\begin{aligned} F_1 &= Z_{0k}\left(\frac{u_1}{i\tan k_k\ell} - \frac{u_2}{i\sin k_k\ell}\right) \\ -F_2 &= Z_{0k}\left(\frac{u_1}{i\sin k_k\ell} - \frac{u_2}{i\tan k_k\ell}\right). \end{aligned} \tag{1.35}$$

Når den elastiske skiven brukes som et koblingslag, blir kraften F_1 utøvd av et piezoelektrisk element. Den virker mot den mekaniske impedansen $Z_{\text{inn}} = F_1/u_1$.

Den elastiske skiven virker videre med kraften F_2 mot utgangsimpedansen $Z_{\text{ut}} = -F_2/u_2$. Ved lydutstråling i vann kan denne settes tilnærmet lik strålingsimpedansen for en plan bølge, $Z_r = F_s/u_2$. F_s er kraften strålingsmediet virker med på koblingslagets overflate.

Fra (1.35) og relasjonen $\frac{1}{\tan a} = \frac{1}{\sin a} - \tan\frac{a}{2}$, finnes da

$$Z_{\text{inn}} = \frac{Z_r + iZ_{0k}\tan k_k\ell}{Z_{0k} + iZ_r\tan k_k\ell} Z_{0k}. \tag{1.36}$$

Dette resultatet er det samme som ligning (1.30).

Kvartbølgeomformer

Impedansene i ligningen over kan divideres med arealet A for å gi det tilsvarende uttrykket for de spesifikke akustiske impedansene,

$$z_r = \rho_r c_r, \quad z_k = \rho_k c_k, \quad z_{\text{inn}} = F_1/(u_1 A).$$

Når $k_k\ell = 2\pi\ell/\lambda$ velges lik et odde antall $\pi/2$, går $\tan k_k\ell$ mot uendelig. Grenseverdien av z_{inn} blir

$$z_{\text{inn}} = \frac{z_k^2}{z_r}.$$

Ved å sette z_k lik $\sqrt{z_r z_m}$, der z_m er den spesifikke akustiske impedansen til det piezokeramiske elementet, oppnås en impedans inn i koblingslaget som er lik transduserelementets egen spesifikke impedans,

$$z_{\text{inn}} = \frac{z_r z_m}{z_r} = z_m.$$

På denne måten kan kvartbølgeomformeren gi en optimal overføring av effekt mellom to medier, så lenge koblingslagets grenseflater er de eneste som gir refleksjoner.

I Masonmodellen er det piezokeramiske elementet en halvbølgeresonator, og det kan vises at de mekaniske seriell- og parallell-Q-faktorene er like når $z_k = (2z_r z_m)^{1/3}$ [13]. Dette skulle gi en optimal akustisk tilpasning [16].

KLM-modellen (Krimholtz, Leedom og Matthaei), som beskriver systemet ved hjelp av en firpolrepresentasjon [12],[6], forutsier at den beste akustiske tilpasningen oppnås når $z_k = (z_r^2 z_m)^{1/3}$. Elementet betraktes i KLM-modellen som sammensatt av to kvartbølgeresonatorer.

Bruk av Mason-modellen

I denne oppgaven ble det valgt å bruke Mason-modellen mer direkte til å velge koblingslagets tykkelse og spesifikke akustiske impedans. Figur 1.7a viser et ekvivalentskjema for et piezokeramisk element med vakuum på baksiden og last på forsiden bestående av et koblingslag og en strålingsimpedans Z_R. Kretsen kan deles i to og omskrives som vist i figur 1.7b og 1.7c, noe som gjør det lettere å beregne transduserens overføringsfunksjon. Ved å kreve at strømfordelingen gjennom de ulike grenene skal være den samme som i krets 1.7a, finnes de nye komponentenes impedanser,

$$
\begin{aligned}
Z_{L1} &= \frac{Z_L Z_{BK}}{Z_L - Z_{AK}} \\
Z_{L2} &= \frac{Z_L(Z_{AK} + Z_R)}{Z_L - Z_{AK}} \\
Z_{M1} &= \frac{Z_A Z_P}{Z_P - (Z_B - \frac{1}{i\omega C_0})} \\
Z_{M2} &= \frac{(Z_A + Z_L) Z_P}{Z_P - (Z_B - \frac{1}{i\omega C_0})}
\end{aligned}
$$

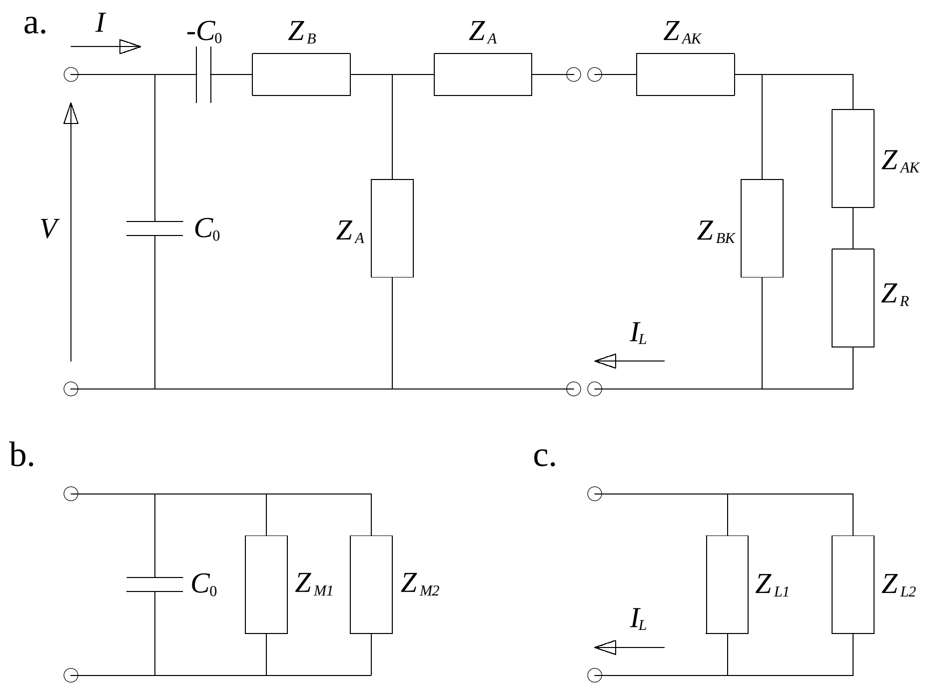
der

$$
\begin{aligned}
Z_L &= \left(\frac{1}{Z_{AK} + Z_R} + \frac{1}{Z_{BK}}\right)^{-1} + Z_{AK}, \\
Z_P &= \left(\frac{1}{Z_A + Z_L} + \frac{1}{Z_A}\right)^{-1} + Z_B - \frac{1}{i\omega C_0}, \\
Z_A &= i\frac{Z_{0m}}{\phi^2} \tan\frac{k_m \ell}{2}, \quad Z_B = \frac{Z_{0m}}{i\phi^2 \sin k_m \ell}, \\
Z_{AK} &= i\frac{Z_{0k}}{\phi^2} \tan\frac{k_k \ell_k}{2}, \quad Z_{BK} = \frac{Z_{0k}}{i\phi^2 \sin k_k \ell_k}, \quad Z_R = \frac{\rho_0 c_0}{\phi^2} A,
\end{aligned}
$$

$Z_{0m} = \rho_m c_m A$, og $Z_{0k} = \rho_k c_k A$. Størrelser med indeks K og k gjelder koblingslaget.

Av ligningene ovenfor kan det ses at kretsens totale admittans er gitt ved

$$
Y_T = i\omega C_0 + 1/Z_P. \tag{1.37}
$$



Figur 1.7: Mason-ekvivalentkrets for et piezokeramisk element med vakuum på baksiden, frontlag og strålingsimpedans Z_R foran koblingslaget.

Sammensetning av koblingslaget

Woods modell (1911) gir lydhastigheten og tettheten i en inhomogen blanding av to faste stoffer, der det ene stoffet forekommer som partikler med tverrsnitt mye mindre enn en bølgelengde [17]. Modellen gjelder når man ser bort fra spredning, viskøse og termiske prosesser, skjærbølger og fasefartens avhengighet av frekvens og partikkelstørrelse.

Den effektive tettheten og kompressibiliteten er gitt ved

$$\rho_e = \rho(1-\varphi) + \rho'\varphi, \quad \kappa_e = \kappa(1-\varphi) + \kappa'\varphi, \tag{1.38}$$

der φ, ρ' og κ' er volumandel, tetthet og kompressibilitet for det "spredte" stoffet, og den kontinuerlige fasen har tetthet og kompressibilitet ρ og κ.

Ligning (1.38) kombinert med $c_e = \sqrt{1/(\rho_e\kappa_e)}$ blir Woods ligning (også kalt Uricks ligning) [17],

$$c_e = \sqrt{\frac{1}{(\kappa(1-\varphi)+\kappa'\varphi)(\rho(1-\varphi)+\rho'\varphi)}}. \tag{1.39}$$

Det kan være praktisk å uttrykke blandingsforholdet ved masseandelen φ_m av den spredte fasen. Sammenhengen mellom volum- og masseandel kan skrives

$$\varphi = \varphi_m \frac{\rho_e}{\rho'} = \frac{\varphi_m \rho}{\rho' + \varphi_m(\rho - \rho')}. \tag{1.40}$$

1.3 Lydforplantning i vann

1.3.1 Grunnleggende ligninger

Utledningen av den ikkelineære bølgeligningen er hentet fra M. F. Hamilton og C. L. Morfeys artikkel i [18]. Ligningen kan utledes av de fire hydrodynamiske grunnligningene. Kontinuitetsligningen og Navier-Stokes' ligning representerer bevaring av henholdsvis masse og bevegelsesmengde,

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{u} = 0, \tag{1.41}$$

$$\rho \frac{D\mathbf{u}}{Dt} + \nabla P = \mu \nabla^2 \mathbf{u} + \left(\mu_B + \frac{1}{3}\mu\right) \nabla \left(\nabla \cdot \mathbf{u}\right). \tag{1.42}$$

I tillegg til disse må entropiligningen og en termodynamisk tilstandsligning tilfredsstilles,

$$\rho T \frac{Ds}{Dt} = \kappa \nabla^2 T + \mu_B \left(\nabla \cdot \mathbf{u}\right)^2 + \frac{1}{2}\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right)^2 \tag{1.43}$$

og

$$P = P(\rho, s). \tag{1.44}$$

I ligningene ovenfor er
$D/Dt = \partial/\partial t + (\mathbf{u} \cdot \nabla)$,
$\nabla^2 = \partial/\partial x + \partial/\partial y + \partial/\partial z$,
ρ tetthet,
t tid,
$\mathbf{u} = (u_x, u_y, u_z)$ partikkelhastighet,
P trykk,
μ skjærviskositet,
μ_B bulkvsikositet,
T temperatur,
s entropi og
κ varmeledningsevne.

1.3.2 Ligning for plane bølger i et tapsfritt medium

Den tapsfrie bølgeligningen $(\nabla^2 - (1/c^2\, \partial^2/\partial t^2)\phi = 0$ kan skrives [19]

$$(\nabla - \frac{1}{c}\frac{\partial}{\partial t}) \cdot (\nabla + \frac{1}{c}\frac{\partial}{\partial t})\phi = 0.$$

Ved å separere bølgeligningen, får man to ligninger som beskriver bølger som forplanter seg i henholdsvis positiv og negativ x-retning,

$$(\nabla + \frac{1}{c}\frac{\partial}{\partial t})\phi^+ = 0, \qquad (\nabla - \frac{1}{c}\frac{\partial}{\partial t})\phi^- = 0.$$

Den endimensjonale bølgeligningen for enkle bølger i positiv x-retning blir dermed

$$(c\frac{\partial}{\partial x} + \frac{\partial}{\partial t})\phi^+ = 0, \tag{1.45}$$

med løsning $\phi^+ = f(x - ct)$.

Kontinuitetsligningen (1.41) og Eulers ligning kan forenkles til en dimensjon og skrives

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} = 0, \tag{1.46}$$

$$\rho\frac{\partial u}{\partial t} + \rho u\frac{\partial u}{\partial x} = -\frac{\partial p}{\partial x} \quad . \tag{1.47}$$

Dersom u er en entydig funksjon av p, slik at $\partial u/\partial t = (du/dp)\partial p/\partial t$ osv., kan (1.46) og (1.47) skrives

$$\frac{d\rho}{dt}\frac{\partial p}{\partial t} + \frac{d(\rho u)}{dp}\frac{\partial p}{\partial x} \quad = \quad 0 \tag{1.48}$$

$$\rho\frac{du}{dp}\frac{\partial p}{\partial t} + (\rho v\frac{du}{dp} + 1)\frac{\partial p}{\partial x} \quad = \quad 0. \tag{1.49}$$

Dette ligningssettet har løsning bare hvis determinanten til koeffisientmatrisen er lik null. Hvis $d\rho/dp = c^{-2}$, får man da at impedansrelasjonen $dv/dp = \pm 1/\rho c$ gjelder for ikkelineære bølger. Ved å velge det positive fortegnet og sette impedansrelasjonen inn i (1.48) eller (1.49), får man

$$\frac{\partial p}{\partial t} + (c+u)\frac{\partial p}{\partial x} = 0. \tag{1.50}$$

En løsning av denne ligningen er $p = f(x - (c+u)t)$, analogt til løsningen av (1.45).

Bølgefarten $(dx/dt)_p = c + u$ kan rekkeutvikles som

$$c + u = c_0 + \tilde{\beta}(0)u + \frac{1}{2}\tilde{\beta}'(0)u^2 + \cdots ,$$

der $\tilde{\beta}(u) = d(c+u)/du$ [18]. For en vilkårlig væske er ikkelinearitetskoeffisienten $\beta = \tilde{\beta}(0) = 1 + B/2A$, der A og B er definert ved

$$\begin{aligned} p = P - P_0 &= \left(\frac{\partial P}{\partial \rho}\right)_{s,0}(\rho - \rho_0) + \frac{1}{2!}\left(\frac{\partial^2 P}{\partial \rho^2}\right)_{s,0}(\rho - \rho_0)^2 + \cdots \\ &= A\left(\frac{\rho'}{\rho_0}\right) + \frac{B}{2!}\left(\frac{\rho'}{\rho_0}\right)^2 + \cdots , \ \rho = \rho_0 + \rho', \end{aligned}$$

dvs. $A = \rho_0(\partial P/\partial \rho)_{s,0} \equiv \rho_0 c_0^2$ og $B = \rho_0^2(\partial^2 P/\partial \rho^2)_{s,0}$.
Ved binomialutvikling av $(c+u)^{-1}$ kan det vises at løsningen av (1.50) til andre orden er lik $q = f(t - x/c_0 + \beta x u/c_0^2)$.

1.3.3 Bølgeligningen til 2. orden

Ligningene (1.41)–(1.43) kan skrives om ved innsetting av $\rho = \rho_0 + \rho'$, $P = P_0 + p$ og $s = s_0 + s'$, der ρ_0, P_0 og s_0 er upertuberte størrelser. Bevegelsesligningen blir videre omformet ved hjelp av vektorrelasjonene $\nabla(\nabla \cdot \mathbf{u}) = \nabla^2\mathbf{u} + \nabla \times \nabla \times \mathbf{u}$ og $(\mathbf{u} \cdot \nabla)\mathbf{u} = \frac{1}{2}\nabla u^2 - \mathbf{u} \times \nabla \times \mathbf{u}$. For lydfelt langt fra vegger kan ledd som inneholder $\nabla \times \mathbf{u}$ neglisjeres, forutsatt at det ikke er strømning i væsken.

De tre ligningene blir til 2. orden henholdsvis

$$\frac{\partial \rho'}{\partial t} + \rho_0 \nabla \cdot \mathbf{u} = -\rho' \nabla \cdot \mathbf{u} - \mathbf{u} \cdot \nabla \rho', \tag{1.51}$$

$$\rho_0 \frac{\partial \mathbf{u}}{\partial t} + \nabla p = \left(\mu_B + \frac{4}{3}\mu\right) \nabla^2 \mathbf{u} - \frac{1}{2}\rho_0 \nabla u^2 - \rho' \frac{\partial \mathbf{u}}{\partial t}, \tag{1.52}$$

$$\rho_0 T_0 \frac{\partial s'}{\partial t} = \kappa \nabla^2 T'. \tag{1.53}$$

Tilstandsligningen kan skrives som en Taylorutvikling av trykket til andre orden,

$$p = c_0^2 \rho' + \frac{c_0^2}{\rho_0} \frac{B}{2A} \rho'^2 + \left(\frac{\partial P}{\partial s}\right)_{\rho',0} s'. \tag{1.54}$$

Ved innsetting av første ordens approksimasjoner i andre ordens ledd, er feilen som gjøres av $O(\tilde{\varepsilon}^3)$. $\tilde{\varepsilon}$ er en ordensparameter av samme orden som mach-tallet. Ligning (1.51) og (1.52) kan derfor skrives som

$$\frac{\partial \rho'}{\partial t} + \rho_0 \nabla \cdot \mathbf{u} = \frac{1}{\rho_0 c_0^4} \frac{\partial p^2}{\partial t} + \frac{1}{c_0^2} \frac{\partial \mathcal{L}}{\partial t}, \tag{1.55}$$

$$\rho_0 \frac{\partial \mathbf{u}}{\partial t} + \nabla p = -\frac{1}{\rho_0 c_0^2} \left(\mu_B + \frac{4}{3}\mu\right) \nabla \frac{\partial p}{\partial t} - \nabla \mathcal{L}, \tag{1.56}$$

der $\mathcal{L} = \frac{1}{2}\rho_0 u^2 - p^2/(2\rho_0 c_0^2)$ er Lagrange-tettheten til andre orden. Ved hjelp av den lineære bølgeligningen $\nabla^2 T' = c_0^{-2}(\partial^2 T')/(\partial t^2)$ og de termodynamiske relasjonene $(\partial P/\partial s)_\rho = \rho^2 (\partial T/\partial \rho)_s$ og $(\partial T/\partial \rho)_s^2 = (c_p - c_v) T c^2/(c_v c_p \rho^2)$ kan (1.53) og (1.54) kombineres til

$$\rho' = \frac{p}{c_0^2} - \frac{1}{\rho_0 c_0^4} \frac{B}{2A} p^2 - \frac{\kappa}{\rho_0 c_0^4} \left(\frac{1}{c_v} - \frac{1}{c_p}\right) \frac{\partial p}{\partial t}. \tag{1.57}$$

Ved å subtrahere den tidsderiverte av ligning (1.55) fra divergensen av (1.56), eliminere ρ' ved hjelp av (1.57) og substituere $\nabla^2 p = c_0^{-2}(\partial^2 p/\partial t^2)$ inn i viskositetsleddet, får man

$$\Box^2 p + \frac{\delta}{c_0^4} \frac{\partial^3 p}{\partial t^3} = -\frac{\beta}{\rho_0 c_0^4} \frac{\partial^2 p^2}{\partial t^2} - \left(\nabla^2 + \frac{1}{c_0^2} \frac{\partial^2}{\partial t^2}\right) \mathcal{L}. \tag{1.58}$$

Dette er den andreordens bølgeligningen. $\Box^2 = \nabla^2 - c_0^{-2}(\partial^2/\partial t^2)$ er d'Alembert-operatoren, og lyd-diffusiviteten δ er gitt ved

$$\delta = \frac{1}{\rho_0}\left(\mu_B + \frac{4}{3}\mu\right) + \frac{\kappa}{\rho_0}\left(\frac{1}{c_v} - \frac{1}{c_p}\right) = \nu\left(\frac{4}{3} + \frac{\mu_B}{\mu} + \frac{\gamma - 1}{\mathrm{Pr}}\right). \tag{1.59}$$

$\nu = \mu/\rho_0$ er her den kinematiske viskositetsparameteren, og γ er forholdet mellom de spesifikke varmekapasitetene c_p og c_v. Prandtls tall er definert som $\mathrm{Pr} = \mu c_p/\kappa$.

1.3.4 Westervelts ligning

Under forutsetning av at $\nabla \times \mathbf{u} = \mathbf{0}$ kan man definere et hastighetspotensial ved $\mathbf{u} = \nabla\phi$. Til første orden blir da $p = -\rho_0 d\phi/dt$, og Lagrange-tettheten kan skrives $\mathcal{L} = \frac{1}{4}\rho_0 \Box^2 \phi^2$. Ved å definere en ny variabel

$$\tilde{p} = p + \frac{\rho_0}{4}\left(\nabla^2 + \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\right)\phi^2 \tag{1.60}$$

og sette inn for p i ligning (1.58), får man

$$\Box^2 \tilde{p} + \frac{\delta}{c_0^4}\frac{\partial^3 \tilde{p}}{\partial t^3} = -\frac{\beta}{\rho_0 c_0^4}\frac{\partial^2 \tilde{p}^2}{\partial t^2}.$$

Det andre leddet på høyre side av ligning (1.60) er stort bare når lokale ikke-lineære effekter er viktige, dvs. for eksempel nær en kilde, før den kumulative forvrengningen har rukket å bygge seg opp. I mange tilfeller gjelder derfor tilnærmingen $\tilde{p} \approx p$, som gir Westervelts ligning,

$$\Box^2 p + \frac{\delta}{c_0^4}\frac{\partial^3 p}{\partial t^3} = -\frac{\beta}{\rho_0 c_0^4}\frac{\partial^2 p^2}{\partial t^2}. \tag{1.61}$$

1.3.5 KZK-ligningen

Den endimensjonale Westerveltligningen

$$\left(\frac{\partial^2}{\partial x^2} - \frac{1}{c_0^2}\frac{\partial^2}{\partial t^2}\right)p + \frac{\delta}{c_0^4}\frac{\partial^3 p}{\partial t^3} = -\frac{\beta}{\rho_0 c_0^4}\frac{\partial^2 p^2}{\partial t^2}$$

har for de to spesialtilfellene $\delta = 0$ og $\beta = 0$ løsningene $p = f(t - x/c_0 + \beta x u/c_0)$ og $p = p_0 \exp(j\omega\tau - \alpha x)$, henholdsvis. Absorpsjonskoeffisienten $\alpha = \delta\omega^2/2c_0^3$, og den retarderte tiden $\tau = t - x/c_0$. I begge tilfellene er koeffisienten foran x av første orden i $\tilde{\varepsilon}$. Løsningene kan skrives på formen $p = p(x_1, \tau)$ der $x_1 = x\tilde{\varepsilon}$. Skrivemåten uttrykker at både ikkelinearitet og absorpsjon gir langsomme variasjoner i lydens forplantningsretning. De to leddene gir bidrag av samme orden.

For en sirkulær plan stempelkilde i planet $z = 0$ defineres Rayleigh-avstanden $r_0 = \frac{1}{2}ka^2$, der $k = \omega/c_0$ er bølgetallet. Fjernfeltet kan defineres ved $z > r_0$, som medfører at $kz > \frac{1}{2}(ka)^2$. Ulikheten antyder at variasjonene på tvers av lydaksen er raskere enn på langs. Dersom diffraksjon inngår i ligningen som et ledd av samme orden som ikkelinearitets- og absorpsjonsleddene, kan følgende såkalte langsomme skala ("slow scale," [18]) benyttes:

$$\begin{aligned}(x_1, y_1, z_1) &= (\tilde{\varepsilon}^{1/2}, \tilde{\varepsilon}^{1/2}, \tilde{\varepsilon}),\\ \tau &= t - x/c_0.\end{aligned}$$

Den transformerte Laplaceoperatoren blir

$$\nabla^2 = \tilde{\varepsilon}\left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial y_1^2}\right) + \tilde{\varepsilon}^2\frac{\partial^2}{\partial z_1^2} - \tilde{\varepsilon}\frac{2}{c_0}\frac{\partial^2}{\partial z_1\,\partial\tau} + \frac{1}{c_0^2}\frac{\partial^2}{\partial\tau^2}.$$

Leddet $\tilde{\varepsilon}^2(\partial^2/\partial z_1^2)$ gir opphav til et tredje ordens ledd i ligningen, og kan neglisjeres. Dette kalles den parabolske approksimasjonen. Innsetting i Westervelts ligning (1.61) gir

$$\tilde{\varepsilon}\left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial y_1^2}\right)p - \tilde{\varepsilon}\frac{2}{c_0}\frac{\partial^2 p}{\partial z_1\,\partial\tau} + \frac{\delta}{c_0^4}\frac{\partial^3 p}{\partial\tau^3} = -\frac{\beta}{\rho_0 c_0^4}\frac{\partial^2 p^2}{\partial\tau^2}.$$

Ved å transformere ligningen tilbake til (x, y, z)-koordinatene får man den parabolske andreordens bølgeligningen, eller KZK-ligningen,

$$\frac{\partial^2 p}{\partial z\,\partial\tau} - \frac{c_0}{2}\nabla_\perp^2 p - \frac{\delta}{2c_0^3}\frac{\partial^3 p}{\partial\tau^3} = \frac{\beta}{2\rho_0 c_0^3}\frac{\partial^2 p^2}{\partial\tau^2}. \tag{1.62}$$

Den transversale Laplaceoperatoren er definert som $\nabla_\perp^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$. Ligningen er oppkalt etter sine opphavsmenn Khokhlov, Zabolotskaya (1969) og Kuznetsov (1971).

1.3.6 Lineært strålingsfelt fra en plan stempelkilde

Feltet langs lydaksen

En plan sirkulær stempelkilde kan beskrives som en sum av enkle kilder med styrke $dQ = U_0\,dS$, montert i et baffel. Det utstrålte feltet fra en enkel kilde kan skrives [1]

$$d\hat{p}(r,t) = j\rho_0 c_0 \frac{dQ\,k}{2\pi r} e^{j(\omega t - kr)}.$$

Feltet fra stempelkilden er gitt ved integralet over kildens overflate S,

$$\hat{p}(r,\theta,t) = j\frac{\rho_0 c_0 U_0 k}{2\pi}\int_S \frac{e^{j(\omega t - k\mathbf{r}')}}{r'}\,dS.$$

$\mathbf{r}'$ er vektoren fra en enkel kilde til observasjonspunktet $\mathbf{r} = (r, \theta)$. Langs lydaksen er $\theta = 0$, og trykkamplituden blir

$$p(r,0) = 2\rho_0 c_0 U_0 \left| \sin\left[\frac{1}{2}kr\left(\sqrt{1+\left(\frac{a}{r}\right)^2} - 1\right)\right]\right|. \tag{1.63}$$

Uttrykket ovenfor oscillerer hurtig nær kilden, for så gradvis å utvikle seg mer som feltet fra en sfærisk kilde. Ekstremalverdiene i amplituden oppnås når

$$\frac{z}{a} = \frac{1}{m}\frac{a}{\lambda} - \frac{m}{4}\frac{\lambda}{a},$$

der $m = 1, 2, 3, \ldots$ og z er avstanden fra kilden målt langs aksen. Når m er odde, oppnår trykkamplituden et maksimum, og når m er jamn, har funksjonen et minimum. Når man beveger seg fra fjernfeltet i retning av kilden, finnes aksetrykkets første maksimum når $m = 1$, det vil si ved

$$z_1 = \frac{a^2}{\lambda} - \frac{\lambda}{4} \approx \frac{a^2}{\lambda}. \tag{1.64}$$

For $z/a \gg 1$ kan rotuttrykket i ligning (1.63) skrives som $1 + \frac{1}{2}(\frac{a}{z})^2$, og hvis i tillegg $z/a \gg ka$, kan (1.63) tilnærmes ved

$$P(r,0) \approx \frac{1}{2}\rho_0 c_0 U_0 \frac{ka^2}{z}.$$

Man ser av dette at når avstanden fra kilden er stor i forhold til kilderadien og bølgelengden, avtar amplituden på aksen tilnærmet på samme måte som i feltet fra en sfærisk kilde.

Feltet på tvers av aksen

I fjernfeltet fra en plan sirkulær stempelkilde med radius a, er trykket gitt ved [1]

$$\mathbf{p}(r,\theta,t) = i\frac{\rho_0 c}{2} U_0 \frac{a}{r} ka e^{-i(\omega t - kr)} \left(\frac{2J_1(ka\sin\theta)}{ka\sin\theta}\right), \tag{1.65}$$

der r er avstanden til kildens midtpunkt og θ er vinkelen fra lydaksen. Faktoren lengst til høyre i uttrykket er den eneste som varierer med θ. Direktivitetsfaktoren $H(\theta)$ er absoluttverdien av denne,

$$H(\theta) = \left|\frac{2J_1(ka\sin\theta)}{ka\sin\theta}\right|. \tag{1.66}$$

$H(\theta)$ er lik 1 for $\theta = 0$, og viser trykkamplitudens variasjon med vinkelen fra lydaksen. J_1 er den førsteordens besselfunksjonen av første type. Funksjonsverdier finnes tabulert i f. eks. Kinsler, Frey, Coppens og Sanders [1].

Vinklene der amplituden er redusert med henholdsvis 3 dB og 6 dB i forhold til på aksen, er gitt ved [4]

$$\theta_{-3\mathrm{dB}} \approx \sin^{-1}\left(\frac{1,616}{ka}\right), \quad \theta_{-6\mathrm{dB}} \approx \sin^{-1}\left(\frac{2,215}{ka}\right). \tag{1.67}$$

Direktivitetsfaktorens første nullpunkt finnes ved

$$\theta_1 \approx \sin^{-1}\left(\frac{3,83}{ka}\right). \tag{1.68}$$

1.4 Numerisk løsning av KZK-ligningen

1.4.1 Koordinattransformasjoner

Endelig-differanse-metoder brukes på en rektangulær grunnmengde. Siden bredden på en lydstråle varierer med posisjonen langs strålens akse, kan koordinattransformasjoner bidra til å spare regnetid.

Ved å innføre dimensjonsløse koordinater, kan ligning (1.62) skrives

$$\frac{\partial^2 p}{\partial\sigma\partial\tau_n} = \frac{1}{4}\nabla^2_{\perp n}\, p + \frac{\delta\omega^2 r_0}{2c_0^3}\frac{\partial^3 p}{\partial\tau_n^3} + \frac{\beta\omega r_0}{2\rho_0 c_0^3}\frac{\partial^2 p^2}{\partial\tau_n^2},$$

der $\sigma = z/r_0$, $r_0 = (1/2)ka^2$, $\tau_n = \omega\tau$ og $\nabla^2_{\perp n} = \partial^2/\partial(x/a)^2 + \partial^2/\partial(y/a)^2 = \partial^2/\partial\xi^2$. a er radien til den sirkulære stempelkilden.

Når absorpsjonskoeffisienten α er mye mindre enn bølgetallet, gjelder tilnærmingen $\alpha \approx \delta\omega^2/2c_0^3$. Sjokkavstanden for en plan og i utgangspunktet sinusformet bølge er gitt ved $l_d = 1/(\beta\varepsilon k)$. Machtallet $\varepsilon = u/c_0 \approx p_0/\rho_0 c_o^2$, der p_0 er bølgens trykkamplitude. Ved å sette inn for δ og β ovenfor, får man

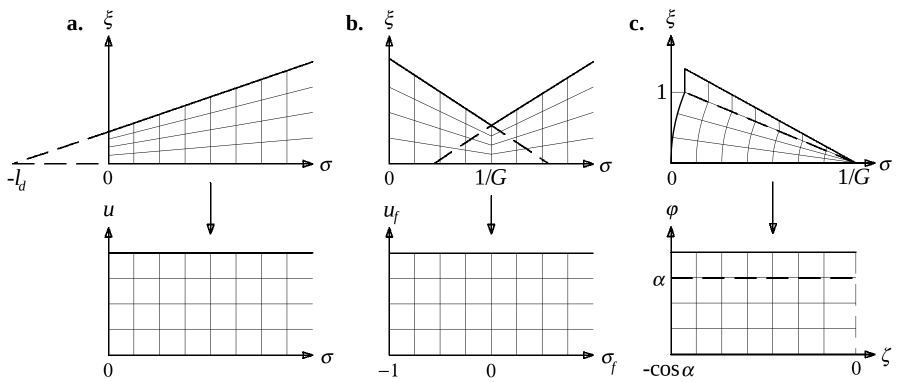
$$\frac{\partial^2 p}{\partial\sigma\partial\tau_n} = \alpha r_0\frac{\partial^3 p}{\partial\tau^3} + \frac{1}{4}\nabla^2_{\perp n}\, p + \frac{r_0}{2l_d p_0}\frac{\partial^2 p^2}{\partial\tau^2}.$$

Trykket kan videre normaliseres til amplituden p_0 ved kilden, slik at man får en dimensjonsløs ligning,

$$\frac{\partial^2 p_n}{\partial\sigma\partial\tau_n} = \alpha r_0\frac{\partial^3 p_n}{\partial\tau^3} + \frac{1}{4}\nabla^2_{\perp n}\, p_n + \frac{r_0}{2l_d}\frac{\partial^2 p_n^2}{\partial\tau^2}, \tag{1.69}$$

der $p_n = p/p_0$. Denne danner utgangspunktet for transformasjonene som er presentert av Berntsen [20] og Naze Tjøtta, Tjøtta og Vefring [21].

Tre transformerte ligninger er nevnt nedenfor. Koordinattransformasjonene er illustrert på figur 1.8.



Figur 1.8: Koordinattransformasjoner for KZK-ligningen. a. TBE (transformert bølgeligning for plane kilder), b. TEFB (transformert bølgeligning for fokuserte stråler), c. MME (mixed model equation)

Transformert bølgeligning for plane og svakt fokuserte kilder (TBE)

Ved simuleringene i denne oppgaven ble den såkalte "transformed beam equation"(TBE) brukt. Transformasjonen

$$\mathbf{u} = \frac{\boldsymbol{\xi}}{1+\sigma}, \quad \tau_p = \tau - \frac{\xi^2}{1+\sigma} \quad T_p = (1+\sigma)p_n \tag{1.70}$$

settes inn i ligning (1.69), slik at man får ([21],[22])

$$\frac{\partial^2 T_p}{\partial\sigma\partial\tau_p} = \alpha r_0 \frac{\partial^3 T_p}{\partial\tau_p^3} + \frac{1}{4(1+\sigma)^2}\nabla^2_{\mathbf{u}} T_p + \frac{r_0}{2l_d(1+\sigma)}\frac{\partial^2 T_p^2}{\partial\tau_p^2}. \tag{1.71}$$

Transformert bølgeligning for fokuserte stråler (TEFB)

Bredden på en fokusert lydstråle varierer på en annen måte enn for en plan kilde. Siden strålen sprer seg sfærisk fra fokalpunktet, velges dette som sentrum i koordinatsystemet, og den longitudinale koordinaten normaliseres til den geometriske fokalavstanden istedenfor Rayleighavstanden,

$$\begin{aligned} \sigma_f &= \frac{z-d}{d} = \sigma G - 1, \quad \mathbf{u}_f = \pm\frac{\boldsymbol{\xi}}{\sigma_f \pm \delta}, \\ \tau_f &= \tau - \frac{r_0}{d}\frac{\xi^2}{\sigma_f \pm \delta}, \quad T_f = (\sigma_f \pm \delta)p_n \end{aligned} \tag{1.72}$$

Gainfaktoren $G = r_0/d$, der d er fokallengden. Det øvre fortegnet gjelder for det postfokale området, $z > d$, mens det nedre gjelder før fokus.

For at koordinatsystemet skal få endelig transversal utstrekning over alt, er koordinatene før og etter fokalplanet sentrert en avstand δ henholdsvis etter - og før fokus. Naze Tjøtta, Tjøtta og Vefring kaller den transformerte ligningen "transformed equation for focused beams" [21],[23],

$$\frac{\partial^2 T_f}{\partial\sigma_f\partial\tau_f} = \alpha d \frac{\partial^3 T_f}{\partial\tau_f^3} + \frac{d}{4r_0(\sigma_f \pm \delta)^2}\nabla^2_{\mathbf{u}_f} T_f + \frac{d}{2l_d(\sigma_f \pm \delta)}\frac{\partial^2 T_f^2}{\partial\tau_f^2}. \tag{1.73}$$

Mixed model equation (MME)

I den innerste delen av nærfeltet til en fokuserende kilde med sfærisk form, kan det være fordelaktig å løse bølgeligningen i et sfærisk koordinatsystem (Naze Tjøtta og Tjøtta [24]). Man antar fortsatt at lydfeltet er symmetrisk om aksen, slik at problemet kan betraktes som todimensjonalt. Avstanden fra det geometriske fokalpunktet brukes som longitudinal variabel, det vil si $\zeta \propto r/d = -z/d\cos\varphi$ når fokalplanet ligger i $z = 0$.

Det sfæriske koordinatsystemet brukes for vinkler $0 \leq \varphi \leq \alpha$, der α er kildens åpningsvinkel. Utenfor dette området er den longitudinale koordinaten lineært forbundet med z. Koordinattransformasjonen er i området mellom kilden og fokus gitt ved [25]

$$\tau_s = \omega(t + \frac{r}{c_0}), \quad p(t,z,\mathbf{x}) = p_0 \frac{d}{r} T_s(\tau_s, \zeta, \varphi, \varepsilon),$$

$$\zeta = \begin{cases} -\dfrac{r\cos\alpha}{d} & \text{for } 0 \le \varphi \le \alpha \\ -\dfrac{r\cos\varphi}{d} & \text{for } \alpha \le \varphi \le \pi/2 \end{cases}. \tag{1.74}$$

Den transformerte ligningen blir

$$\frac{\partial^2 T_s}{\partial \tau_s \partial \zeta} = -\frac{C_{diff}}{2kd\zeta^2} D_{\varphi,\varepsilon} T_s - \alpha d C_{abs} \frac{\partial^3 T_s}{\partial \tau_s^3} + \frac{d}{2l_d} \frac{1}{\|\zeta\|} \frac{\partial^2 T_s^2}{\partial \tau_s^2}, \tag{1.75}$$

der

$$\begin{aligned} C_{diff} &= \begin{cases} -\cos\alpha & \text{for } 0 \le \varphi \le \alpha \\ -\cos\varphi & \text{for } \alpha \le \varphi \le \pi/2 \end{cases}, \\ C_{abs} &= \begin{cases} -\frac{1}{\cos\alpha} & \text{for } 0 \le \varphi \le \alpha \\ -\frac{1}{\cos\varphi} & \text{for } \alpha \le \varphi \le \pi/2 \end{cases}, \\ D_{\varphi,\varepsilon} &= \frac{1}{\sin\varphi} \frac{\partial}{\partial\varphi}(\sin\varphi \frac{\partial}{\partial\varphi}) + \frac{1}{\sin^2\varphi} \frac{\partial^2}{\partial\varepsilon^2}. \end{aligned}$$

1.4.2 Fouriertransformasjon

Ligningene ovenfor kan Fourier-dekomponeres ved innsetting av

$$T = \sum_{n=1}^{\infty} (a_n \sin n\tau + b_n \cos n\tau), \tag{1.76}$$

der a_n og b_n er avhengige av de romlige koordinatene. Ved innsetting i den dimensjonsløse bølgeligningen (1.69) får man [26],[20],[27]

$$\begin{aligned} \frac{\partial a_n}{\partial \sigma} &= -\alpha r_0 n^2 a_n + \frac{1}{4n} \nabla_\perp^2 b_n + \frac{r_0 n}{2l_d} \Bigg(\frac{1}{2} \sum_{i=1}^{n-1} (a_i a_{n-i} - b_i b_{n-i}) \\ &\quad - \sum_{i=n+1}^{\infty} (a_i a_{i-n} + b_i bi - n) \Bigg), \\ \frac{\partial b_n}{\partial \sigma} &= -\alpha r_0 n^2 b_n - \frac{1}{4n} \nabla_\perp^2 a_n + \frac{r_0 n}{2l_d} \Bigg(\sum_{i=1}^{n-1} (a_i b_{n-i}) \\ &\quad + \sum_{i=n+1}^{\infty} (a_i b_{i-n} - b_i a_{i-n}) \Bigg), n = 1, 2, \ldots. \end{aligned} \tag{1.77}$$

Ligningene (1.71), (1.73) og (1.75) kan også transformeres ved hjelp av (1.76), og gir ligningssett i frekvensdomenet på samme form som (1.77). Alle de ovennevnte versjonene av KZK-ligningen kan derfor løses med den samme numeriske algoritmen.

Ifølge Naze Tjøtta et al. [26] kan spektralformen av bølgeligningen generaliseres for å ta hensyn til dispersjon og mer generelle absorpsjonslover. Dette gjøres ved å erstatte det første leddet på høyre side i hver av ligningene (1.77) med lineære kombinasjoner av a_n og b_n.

Løsningen av ligningssettet tar utgangspunkt i en kildebetingelse i planet $\sigma = \sigma_{start}$ på formen $p_n = f(\xi, \tau)$. Kildebetingelsen for a_n og b_n blir da [26]

$$\begin{aligned} a_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(\xi,\tau)\sin n\tau d\tau, \\ b_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(\xi,\tau)\cos n\tau d\tau. \end{aligned} \tag{1.78}$$

1.4.3 Endelig differanse-metoder

KZK-ligningen er parabolsk og kan løses med endelig differanse-metoder.

Regneområdet $0 \leq u \leq u_{maks}$, $\sigma_{start} \leq \sigma \leq \sigma_{slutt}$, deles i $I \times J$ like store ruter, slik at

$$\begin{aligned} u_j &= j \cdot \Delta u, & j &= 0, 1, 2, \ldots, J, \\ \sigma_i &= i \cdot \Delta\sigma + \sigma_{start}, & i &= 0, 1, 2, \ldots, I. \end{aligned}$$

I tillegg til ligninger og kildebetingelser tilsvarende ligning (1.77) og (1.78), trenges to randbetingelser. Betingelsen for $u = 0$ er

$$\left(\frac{\partial a_n}{\partial u}\right)_{u=0} = 0, \quad \left(\frac{\partial b_n}{\partial u}\right)_{u=0} = 0$$

fordi løsningen skal være symmetrisk om aksen.

Refleksjoner fra randen $u = u_{maks}$ gjør løsningen ustabil, men kan som oftest unngås ved at u_{maks} blir valgt stor nok. I Ystad og Berntsens versjon av Bergenkoden [23] dempes løsningen eksponensielt ut mot randen for å redusere refleksjonene.

For $\sigma = \sigma_{slutt}$ er det ikke nødvendig å definere noen randbetingelse. I hvert regnesteg beregnes løsningen for σ_i på grunnlag av forrige steg, dvs. funksjonsverdiene for σ_{i-1}.

Differensoperatorer

Differensialene i ligningssettet erstattes med endelige differanser [28]. I transversal retning brukes en trepunkts differanse gyldig til annen orden,

$$\left(\frac{\partial^2 g}{\partial u^2}\right)_{u_j,\sigma_i} = \frac{g_{j+1}^i - 2g_j^i + g_{j-1}^i}{(\Delta u)^2} + 0((\Delta u)^2), \tag{1.79}$$

der g_j^i er løsningen i punktet (σ_i, u_j).

I σ-retning brukes tre ulike differenskvotienter.

$$\begin{aligned}\left(\frac{\partial g}{\partial \sigma}\right)_{u_j,\sigma_i} &= \frac{g_j^i - g_j^{i-1}}{\Delta\sigma} + O((\Delta\sigma)^2),\\ \left(\frac{\partial g}{\partial \sigma}\right)_{u_j,\sigma_i} &= \frac{g_j^{i-1} - g_j^i}{\Delta\sigma} + O((\Delta\sigma)^2)\end{aligned} \tag{1.80}$$

er henholdsvis en bakover- og en foroverdifferanse.

Når det brukes foroverdifferanse i den longitudinale retningen, er løsningen i hvert punkt (u_j, σ_i) gitt ved tre funksjonsverdier fra forrige regnesteg, nemlig løsningen i punktene $(u_{j\pm1}, \sigma_{i-1})$ og (u_j, σ_{i-1}). Denne algoritmen kalles eksplisitt fordi løsningen i ethvert punkt i planet $\sigma = \sigma_i$ kan regnes ut direkte. En ulempe med den eksplisitte metoden er at den kan bli ustabil hvis ikke forholdet $\Delta\sigma/n(\Delta u)^2$ er lite nok.

I Bergenkoden brukes også en variant av Crank-Nicolson-metoden. En topunkts differanse i σ-retning oppnås ved å tolke den andre av ligning (1.80) som den deriverte i punktet midt mellom (u_j, σ_{i+1}) og (u_j, σ_i). Den tilsvarende annenderiverte i transversal retning settes lik gjennomsnittet av differansekvotientene for $\sigma = \sigma_{i+1}$ og $\sigma = \sigma_i$,

$$\begin{aligned}\left(\frac{\partial g}{\partial \sigma}\right)_{u_j,(\sigma_{i+1}+\sigma_i)/2} &= \frac{g_j^{i+1} - g_j^i}{\Delta\sigma},\\ \left(\frac{\partial^2 g}{\partial u^2}\right)_{u_j,(\sigma_{i+1}+\sigma_i)/2} &= \frac{1}{2}\left(\frac{g_{j+1}^{i+1} - 2g_j^{i+1} + g_{j-1}^{i+1}}{(\Delta u)^2} + \frac{g_{j+1}^i - 2g_j^i + g_{j-1}^i}{(\Delta u)^2}\right).\end{aligned} \tag{1.81}$$

Ved å bruke Crank-Nicolson-metoden oppnår man andreordens differanser i både transversal og longitudinal retning. Dette gjør løsningsmetoden stabil uansett valg av steglengder, selv om nøyaktigheten er best for små $\Delta\sigma/n(\Delta u)^2$. Crank-Nicolson-metoden er implisitt, noe som betyr at løsningen i et punkt (u_j, σ_i) ikke er gitt direkte ved resultatet fra forrige regnesteg. I tillegg til funksjonsverdien i punktet (u_j, σ_{i-1}) trenges også løsningene i (u_{j+1}, σ_i) og (u_{j-1}, σ_i). Det må derfor løses et sett med simultane ligninger i hvert steg, dvs. for hver σ_i. Ligningene (1.77) gir tridiagonale koeffisientmatriser, slik at problemet kan løses ved tilbakesubstitusjon.

1.4.4 Bergenkoden

Den numeriske løsningen av den transformerte bølgeligningen (1.71) er implementert av Jarle Berntsen og Erlend Vefring [29]. En tidligere versjon er blitt omtalt av Aanonsen et al. [30], men brukes ikke her. Løsningene gjelder for problemer med rotasjonssymmetri om z-aksen, som lydstrålen forplanter seg langs.

Når ligningssettet løses numerisk, trunkeres summene i ikkelinearitetsleddene. Dette gir en opphopning av energi i de høyeste harmoniske som er tatt med. Feilen forplanter seg til alle frekvenskomponentene i løsningen, men forblir viktigst for de høyeste harmoniske. Når man tar med så mange harmoniske i utregningen at en økning på en i antallet ikke gir noen nevneverdig endring av resultatet, antas det i denne oppgaven at trunkeringsfeilen er neglisjerbar.

I Bergenkoden brukes hovedsakelig Richtmyer-prosedyren, som er en variant av Crank-Nicholson-metoden. Metoden har den ulempen at diskontinuiteter i kildebetingelsen forårsaker Gibbske oscillasjoner i løsningen. For å unngå dette problemet er det brukt en annen algoritme i de første få regnestegene. Aanonsen-metoden benytter rene forover- og bakoverdifferanser, og gir en dempning av løsningen. Dempningen virker sterkest på de høyere harmoniske.

Kapittel 2

Eksperiment

2.1 Spesifikasjoner for lydkilden

Transduserens driftsfrekvens var ønsket å ligge rundt 1 MHz. Tanken der SOBER-eksperimentet skulle utføres, var 50 cm bred, 70 cm høy og 270 cm lang. Det skulle måles på en lydstråle i tankens lengderetning, med minst mulig refleksjoner fra sideveggene. Det var planlagt å reflektere lydstrålen en gang i bunnen av tanken, slik at lengden av lydstrålen som det skulle måles i, kunne komme opp i 3 m. Hvis hovedloben skulle være smalere enn 50 cm ved denne avstanden, måtte åpningsvinkelen være mindre enn 9,5°. Det ble valgt å bruke en piezokeramisk skive med diameter 5 cm, som skulle tilfredsstille kravet om direktivitet med god margin.

For å hindre uønskede refleksjoner i å påvirke det målte lydfeltet, ble det planlagt å bruke pulset lyd. Ved målinger nær grenseflater var det ønskelig å gjøere lydpulsene kortest mulig. I denne sammenhengen var det viktig at også lydkildens transientrespons var kort.

Transduserens vertikale posisjon og vinkel skulle justeres underveis i løpet av SOBER-eksperimentet. Tiltingen måtte kunne gjøres med en nøyaktighet på 0,1 grad. Da lydkilden skulle plasseres inne i det stratifiserte mediet, der det var viktig å unngå unødig omrøring, var det også nødvendig å gjøre den mer strømlinjeformet enn transduserne som var tilgjengelige fra tidligere eksperiment. Det ble også forsøkt å konstruere kilden så kompakt som mulig.

Siden kildens oppgave var å generere ikkelineære lydfelt, måtte den være i stand til å operere med høy utgangseffekt. Ønsket om høy amplitude fikk blant annet betydning for valget av piezoelektrisk materiale.

Tabell 2.1: Skivenes dimensjoner.

Skive	Diameter		Tykkelse	
	$\bar{d}$ [mm]	s_d [mm]	$\bar{\ell}$ [mm]	s_ℓ [mm]
5A	49,977	0,013	2,056	0,0015
5B	49,976	0,014	2,045	0,0020
5C	49,966	0,0084	2,032	0,0021
5D	49,955	0,0053	2,032	0,0018
5E	49,978	0,0092	2,030	0,0021
5F	49,974	0,012	2,048	0,0018

2.2 Målinger og simuleringer av piezokeramiske skiver i luft

Det ble målt på seks skiver av det piezokeramiske materialet Pz 26 fra Ferroperm, med tykkelse 2 mm og diameter 5 cm. Tre av dem (5A, 5B og 5C) var blitt brukt tidligere, mens resten (5D, 5E og 5F) ble kjøpt nye til SOBER-eksperimentet. Pz 26 ble valgt fordi det er et såkalt hardt piezokeramisk materiale, som tåler å operere med store amplituder.

2.2.1 Bestemmelse av materialkonstanter

Diameter og tykkelse ble målt ti tilfeldige steder på hver skive med henholdsvis et TESA digit-cal SI skyvelær og en TESA digit 0–25 mm mikrometerskrue. Tabell 2.1 viser middelverdier og standardavvik for målingene.

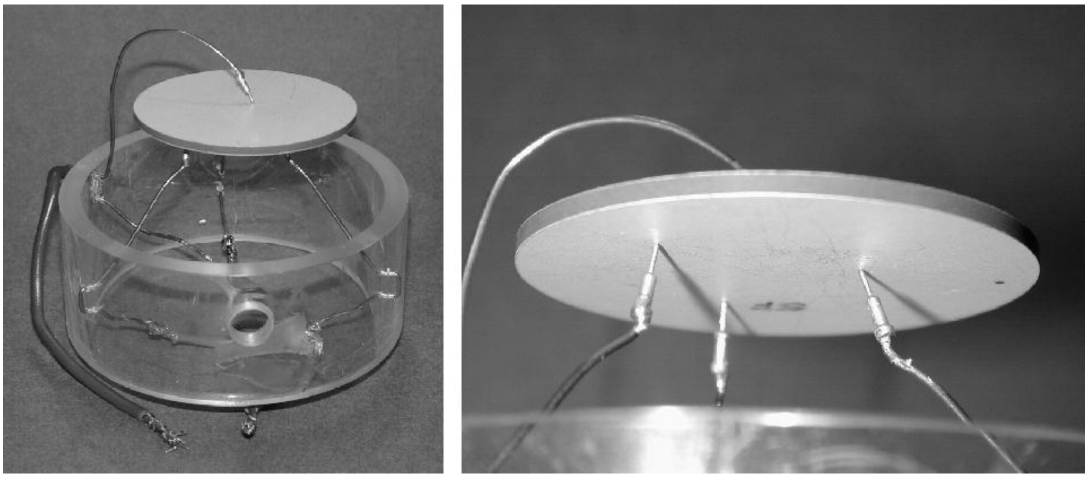
For hvert av elementene ble fem karakteristiske frekvenser bestemt ved hjelp av en Hewlett Packard 4192A impedansanalysator med HP 16047A tilkoblingsmodul. Resonansfrekvensene påvirkes av mange faktorer, som for eksempel temperatur, lufttrykk, luftfuktighet og den mekaniske påvirkningen fra måleinstrumenter [31].

Elementene ble plassert i holderen i figur 2.1, der tre ledere fra undersiden og en fra oversiden sørget de for den elektriske kontakten mellom elementet og impedansanalysatoren. Samtidig holdt de den piezokeramiske skiven på plass.

Den relative luftfuktigheten ved målingene var omkring 50%, og temperaturen lå på (23 ± 1)°C. Dersom temperaturvariasjonene for Pz26 svarer til Rune Fardals resultater for PZT-5A [31], blir resonansfrekvensene med ubestemtheter som i tabell 2.2.

Med impedansanalysatoren ble også elementenes totale kapasitanser C_{0T} målt ved 400 Hz. Frekvensen for kapasitansmåling ble valgt mindre enn 1/100 av skivenes første serieresonans, som anbefalt av Frode Atterås [32].

På grunnlag av måledataene ble noen av materialkonstantene (1.7) bestemt. Dette ble gjort ved hjelp av relasjoner som bygger på Masonmodellen, oppsummert i [32] og [33]. Til utregningen ble det brukt en av Atterås' Matlab-filer,



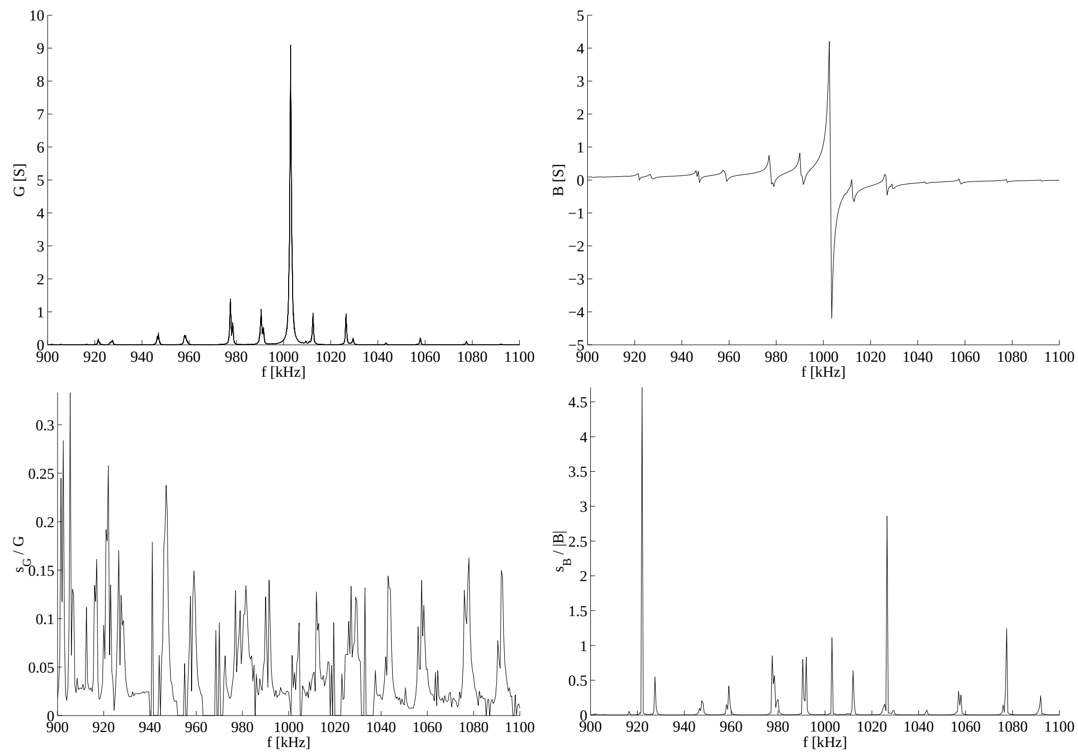
Figur 2.1: Holder for måling av impedans i en piezoelektrisk skive.

`skive.m`, som er listet i hans hovedfagsoppgave [32]. De beregnede parameterne for skive 5D og 5E er satt opp sammen med produsentens oppgitte størrelser [34] i tabell 2.3. Ved utregningene ble den oppgitte verdien brukt for massetettheten, og oppgitte verdier for s_{33}^E og d_{33} er satt inn ved utregning av tallene i parentes. Ferroperms data er gitt med relative usikkerheter på 10% for elektriske -, 5% for elektromekaniske - og 2,5% for mekaniske egenskaper.

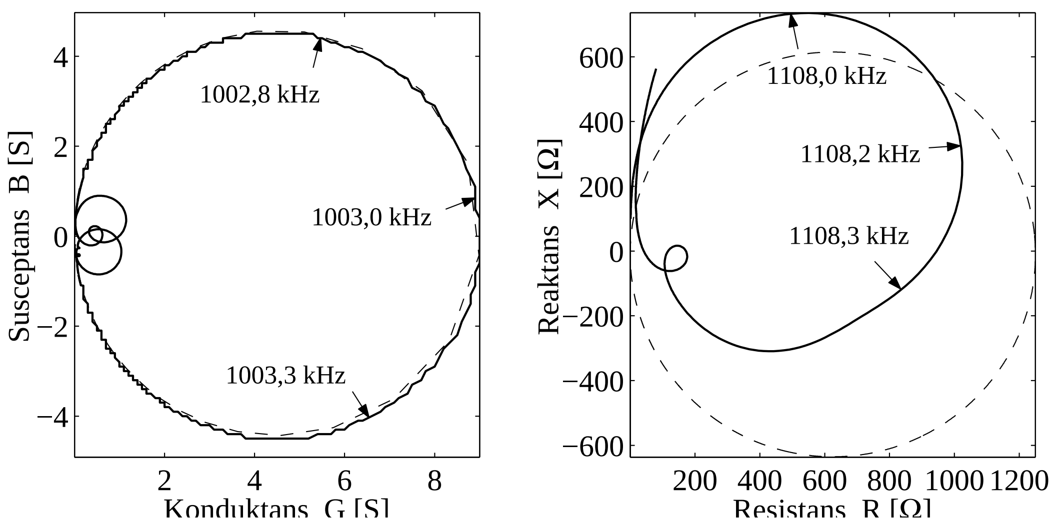
2.2.2 Frekvensrespons i luft

Admittansen ble målt ti ganger for hver skive. Før hver måling ble elementet satt i holderen på nytt, og den mekaniske spenningen i holderen ble endret. Dette ble gjort for å oppnå en størst mulig variasjon i hvordan holderen påvirket elementet. Resultatet av de ti målingene på skive 5D er plottet i figur 2.2. Variasjonene ved måling på de andre skivene var tilsvarende.

I figur 2.3 er målt admittans og impedans for skive 5D sammenlignet med utregnede verdier basert på Mason-modellen (1.27),(1.28). Det venstre plottet viser admittansen for frekvenser rundt tykkelses-serieresonans, mens det høyre er av impedansen rundt tykkelses-parallellresonans. Den sistnevnte frekvensen var vanskelig å finne fordi resistansen hadde flere store maksima i det aktuelle frekvensområdet.



Figur 2.2: Ti målinger av konduktans og susceptans for skive 5D, standardavvik relativt til middelverdiene.

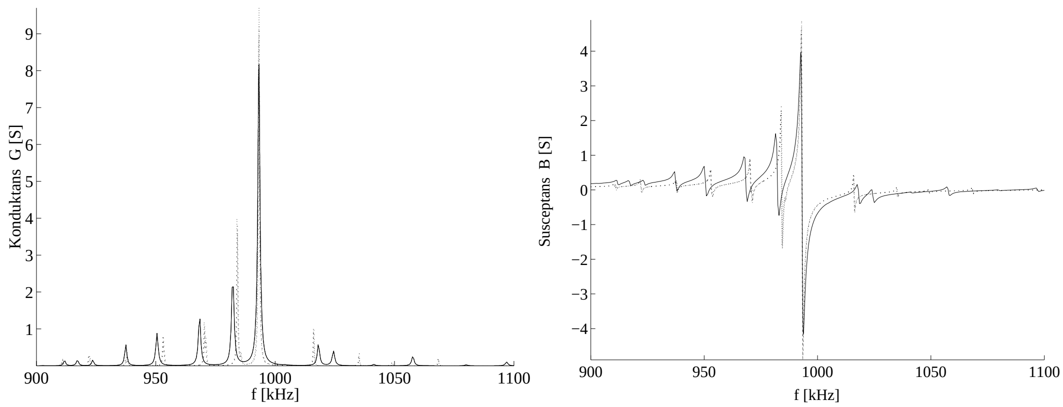


Figur 2.3: Skive 5D, til venstre: Admittans (GB-plott). Til høyre: Impedans (RX-plott). Heltrukken kurve viser måleresultat, mens stiplet kurve er utregnet ved hjelp av Mason-modellen.

Tabell 2.2: Karakteristiske frekvenser for skivene ved (23±1)°C: Første og andre radielle serieresonans, første radielle parallellresonans, første tykkelses-serieresonans og første tykkelses-parallellresonans. Kapasitansene ble målt ved 400 Hz.

Skive	1. radiell serie f_{1r} [kHz]	2. radiell serie f_{2r} [kHz]	1. radiell parallell f_p [kHz]
5A	43,98 ± 0,04	114,9 ± 0,1	50,65 ± 0,02
5B	44,14 ± 0,04	115,1 ± 0,1	50,58 ± 0,02
5C	44,31 ± 0,04	115,6 ± 0,1	50,52 ± 0,02
5D	44,12 ± 0,04	115,2 ± 0,1	53,03 ± 0,02
5E	44,07 ± 0,04	115,1 ± 0,1	50,67 ± 0,02
5F	43,93 ± 0,04	114,7 ± 0,1	50,70 ± 0,02

Skive	1. tykkelse serie f_{st} [kHz]	1. tykkelse parallell f_{pt} [kHz]	Kapasitans C_{0T} [nF]
5A	993,2 ± 0,1		11,8 ± 0,4
5B	1000,1 ± 0,1	1105,36 ± 0,08	12,0 ± 0,4
5C	1005,2 ± 0,1	1110,88 ± 0,08	12,0 ± 0,4
5D	1003,1 ± 0,1	1108,29 ± 0,08	11,8 ± 0,4
5E	1005,2 ± 0,1	1110,83 ± 0,08	11,9 ± 0,4
5F	998,0 ± 0,1	1103,21 ± 0,08	11,9 ± 0,4



Figur 2.4: Konduktans og susceptans for skive 5A. Den heltrukne kurven viser simulerte verdier fra FEMP, mens de stiplede er måleresultater.

Tabell 2.3: Materialparametere for Ferroperm Pz 26 — oppgitt fra Ferroperm [34] og beregnet ved hjelp av Mason-modellen [32].

Parameter		Oppgitt	Mason, 5D	Mason, 5E
c_{11}^E	$[10^{11}\ \mathrm{N/m^2}]$	1,68±0,04	(1,7 ±0,2)	(1,7 ±0,1)
c_{12}^E	$[10^{11}\ \mathrm{N/m^2}]$	1,10±0,03	(1,1 ±0,1)	(1,1 ±0,1)
c_{13}^E	$[10^{11}\ \mathrm{N/m^2}]$	1,00±0,03	(1,0 ±0,1)	(1,0 ±0,1)
c_{33}^E	$[10^{11}\ \mathrm{N/m^2}]$	1,23±0,03	1,229±0,005	1,231±0,005
c_{33}^D	$[10^{11}\ \mathrm{N/m^2}]$	1,58±0,04	1,562±0,006	1,566±0,006
c_{44}^E	$[10^{10}\ \mathrm{N/m^2}]$	3,01±0,08		
c_{66}^E	$[10^{10}\ \mathrm{N/m^2}]$	2,88±0,07		
s_{33}^E	$[10^{-11}\ \mathrm{m^2/N}]$	1,96±0,05		
d_{33}	$[10^{-10}\ \mathrm{C/N}]$	3,3 ±0,2		
e_{31}	$[\mathrm{C/m^2}]$	-2,0 ±0,1	(-8 ±6)	(-3 ±6)
e_{33}	$[\mathrm{C/m^2}]$	14,7 ±0,7	14,0 ±0,3	14,9 ±0,3
e_{15}	$[\mathrm{C/m^2}]$	9,9 ±0,5		
ε_{11}^S	$[10^{-9}\ \mathrm{C/Vm}]$	7,4 ±0,7		
ε_{33}^S	$[10^{-9}\ \mathrm{C/Vm}]$	6,2 ±0,6	5,8 ±0,2	6,7 ±0,2
ρ	$[10^3\ \mathrm{kg/m^3}]$	7,7 ±0,2		
Q_m		780 ±20		
$\tan\delta_e$	$[10^{-3}]$	3,0 ±0,3		

2.2.3 FEM-simulering av svingemønster

Programmet FEMP 3.0 ble skrevet av Jan Kocbach [16],[35], og bruker endelig element-metode til simulering av svingninger i piezoelektriske materialer, elastiske materialer og væsker.

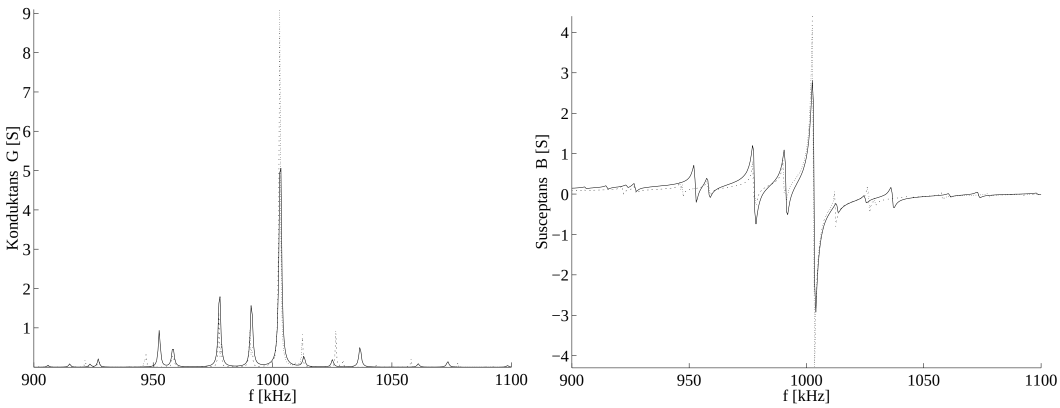
Ved bruk av de oppgitte materialparameterne fra Ferroperm ble det observert dårlig overensstemmelse mellom målte og simulerte admittanskurver. Materialkonstantene som ble funnet ved hjelp av Mason-modellen ble satt inn i modellen, og nye verdier for de resterende parameterne ble funnet ved å forsøke å tilpasse admittanskurver fra simuleringsprogrammet til de målte dataene (tabell 2.4). Simulerte og målte admittanser er vist i figur 2.4 og 2.5.

De simulerte svingemønsterne rundt første tykkelses-serieresonans (TE-moden), med materialparametere fra tabell 2.4, er illustrert i figur 2.6 og figur 2.7. Sammenligning med svingemønsteret for en skive med Ferroperms oppgitte parametere, viste at egenskapene til de to skivene ikke var blitt bestemt på riktig måte. De oppgitte parameterne førte til tydelige utslag i tykkelsesretning (figur 2.8), mens dataene som fremkom ved tilpasning av admittanskurver ga helt andre bevegelsesmønster i frekvensområdet rundt den antatte TE-moden. Partikkelforskyvningen på figurene er multiplisert med en faktor 5 for å gjøre detaljene bedre synlige. Dette har medført at elementer overlapper på noen av figurene, noe som er ufysisk og ikke et egentlig resultat av simuleringene.

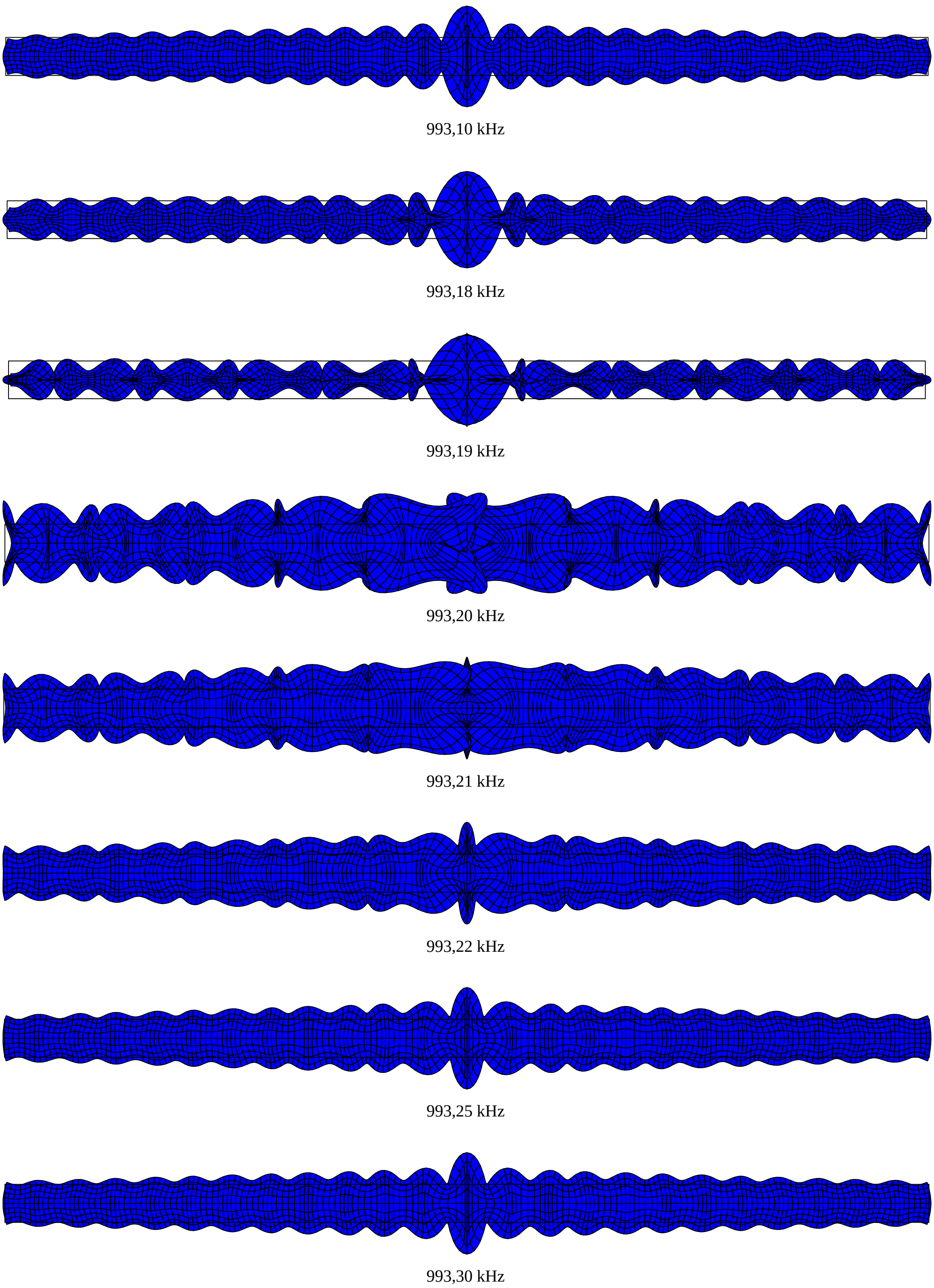
Den opprinnelige planen var å simulere større deler av transduserkonstruksjonen, men dette ble gjort vanskelig av geometrien og metodens langsomme konvergens for det aktuelle problemet. Det ble derfor valgt å ikke arbeide videre med bestemmelse av materialparametere.

Tabell 2.4: Materialparametere brukt i FEMP for å simulere svingningene i skive 5A og skive 5D.

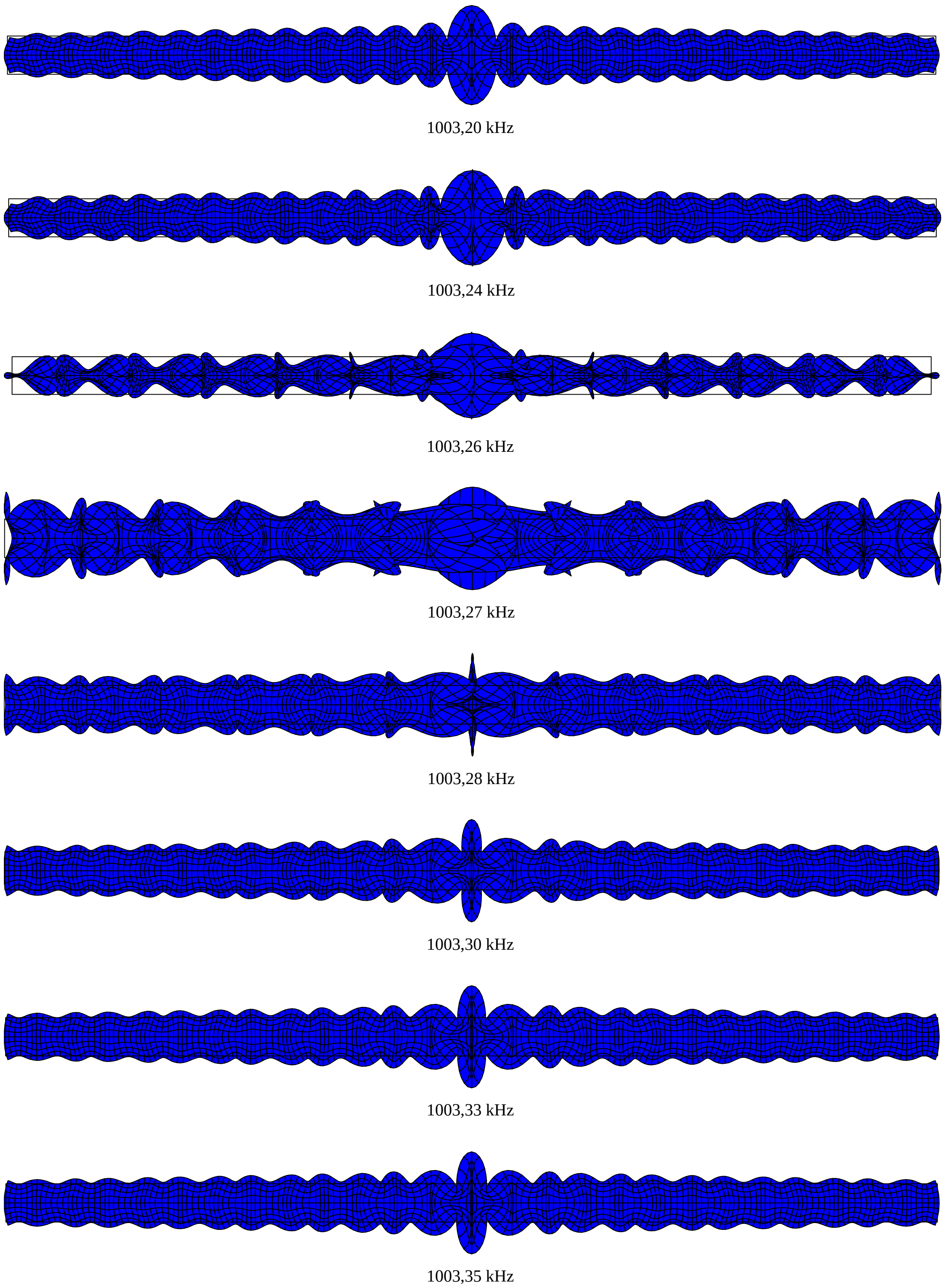
Parameter		Skive 5A	Skive 5D
c_{11}^E	[N/m^2]	$1{,}7180 \cdot 10^{11}$	$1{,}4600 \cdot 10^{11}$
c_{12}^E	[N/m^2]	$1{,}1105 \cdot 10^{11}$	$1{,}0784 \cdot 10^{11}$
c_{13}^E	[N/m^2]	$9{,}7200 \cdot 10^{10}$	$9{,}9775 \cdot 10^{10}$
c_{33}^E	[N/m^2]	$1{,}2120 \cdot 10^{11}$	$1{,}2350 \cdot 10^{11}$
c_{44}^E	[N/m^2]	$2{,}8550 \cdot 10^{10}$	$3{,}0100 \cdot 10^{10}$
c_{66}^E	[N/m^2]	$2{,}8800 \cdot 10^{10}$	$2{,}8800 \cdot 10^{10}$
e_{31}	[C/m^2]	-3,248	-3,700
e_{33}	[C/m^2]	15,16	13,8526
e_{15}	[C/m^2]	9,860	9,900
ε_{11}^S	[C/Vm]	$7{,}3580 \cdot 10^{-9}$	$7{,}3580 \cdot 10^{-9}$
ε_{33}^S	[C/Vm]	$6{,}6800 \cdot 10^{-9}$	$5{,}8338 \cdot 10^{-9}$
ρ	[kg/m^3]	7700	7700
Q_m		1000	1060
$\tan \delta_e$		0,003	0,003



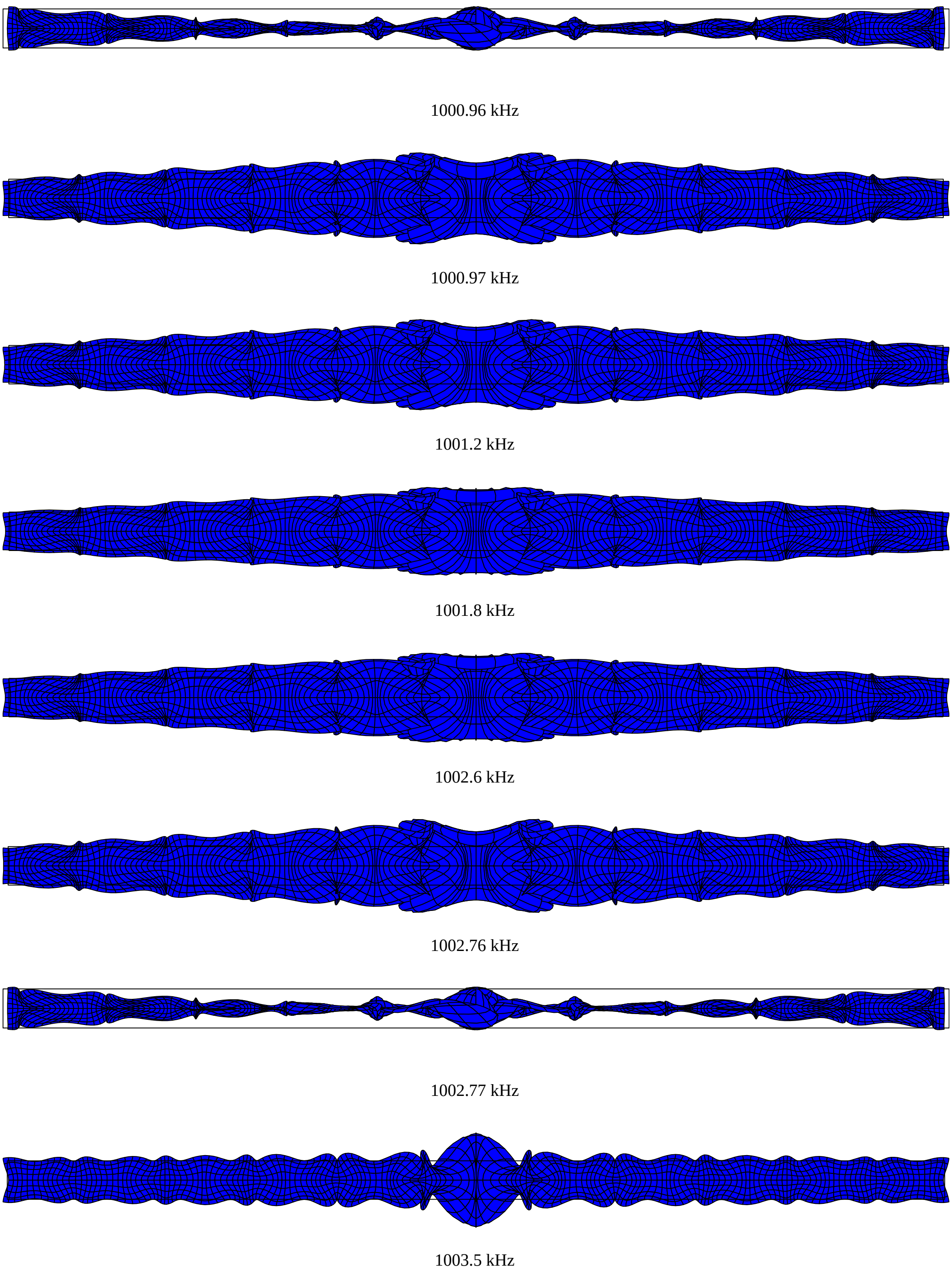
Figur 2.5: Konduktans og susceptans for skive 5D. Den heltrukne kurven viser simulerte verdier fra FEMP, mens de stiplede er måleresultater.



Figur 2.6: Simulert svingemønster i vakuum for skive 5A. Tegningene viser relative forskyvningsamplituder i skiven ved ulike frekvenser omkring serieresonans. De beregnede forskyvningsamplitudene er multiplisert med en faktor for å bli bedre synlige. Dette medfører at noen av elementene overlapper, noe som ikke er et egentlig resultat av simuleringene.



Figur 2.7: Simulert svingemønster i vakuum for skive 5D. Tegningene viser relative forskyvningsamplituder i skiven ved ulike frekvenser omkring serieresonans. De beregnede forskyvningsamplitudene er multiplisert med en faktor for å bli bedre synlige. Dette medfører at noen av elementene overlapper, noe som ikke er et egentlig resultat av simuleringene.



Figur 2.8: Simulert svingemønster i vakuum, oppgitte parametere. Tegningene viser relative forskyvningsamplituder i skiven ved ulike frekvenser omkring serieresonans. De beregnede forskyvningsamplitudene er multiplisert med en faktor for å bli bedre synlige. Dette medfører at noen av elementene overlapper, noe som ikke er et egentlig resultat av simuleringene.

2.3 Simuleringer av mekaniske koblingslag

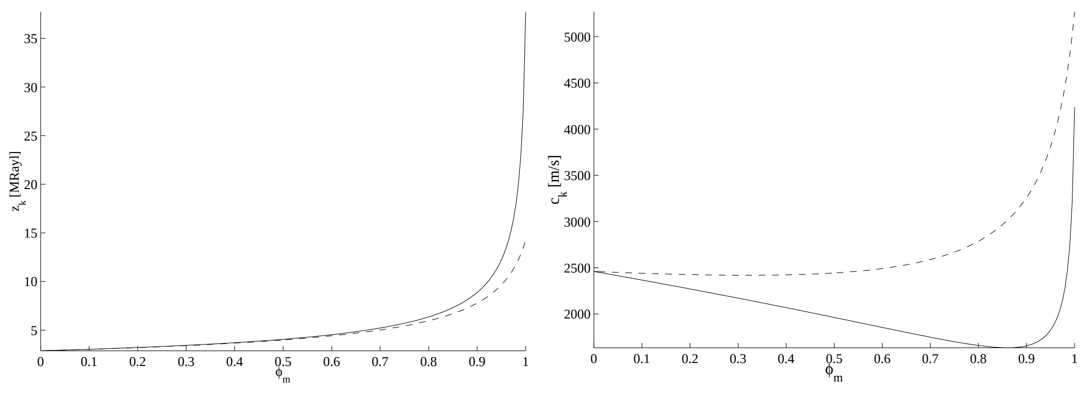
Transduserens båndbredde ble utvidet ved å plassere et koblingslag mellom det piezokeramiske elementet og strålingsmediet. Laget ble støpt direkte på elementet, og besto av en blanding av epoksid-harpiks og kobberpulver med maksimal kornstørrelse 63μm. Aluminiumholdige frontlag og et uten tilsatt metall ble også vurdert.

Som epoxymateriale ble det brukt Araldite AY 103 og herder HY 956, i blandingsforhold fem massedeler til en.

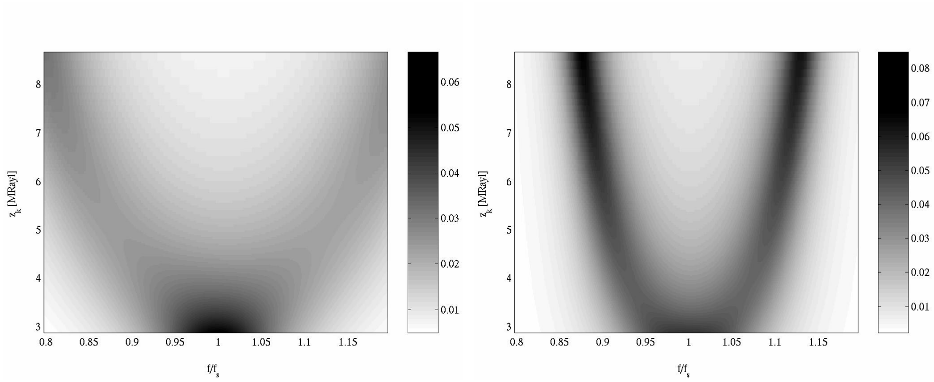
Ved innsetting av verdiene fra tabell 2.5 i ligningene (1.38)–(1.40), ble det oppnådd en teoretisk sammenheng mellom den spesifikke akustiske impedansen z_k i koblingslaget og masseandelen av metallpulver φ_m. Resultatet er plottet i figur 2.9. Her er også lydfartens avhengighet av blandingsforholdet tatt med. De heltrukne kurvene viser verdier for Araldite og kobber, mens de stiplede er for blandinger av Araldite og aluminiumpulver.

Tabell 2.5: Egenskaper for materialer brukt i koblingslagene. Tetthet og lydfart for AY 103 ble målt, mens verdiene for aluminium og kobber er hentet fra en tabell av Laust Pedersen [36].

	Tetthet ρ [kg/dm^3]	Lydfart c_l [km/s]	Impedans z_m [MRayl]
AY 103/HY 956	1,17$\pm$0,01	2,46$\pm$0,02	2,88$\pm$0,03
Aluminium	2,7	6,32	17
Kobber	8,9	5,01	45



Figur 2.9: Til venstre: Spesifikk akustisk impedans som funksjon av masseandelen av metall i blandingen med Araldite 103/956. Til høyre: Lydfarten i koblingslaget som funksjon av det samme blandingsforholdet. De stiplede kurvene viser verdier for aluminiumpulver, mens de heltrukne er for kobber.



Figur 2.10: Konduktans i Siemens i elementet som funksjon av frekvens og koblingslagets spesifikke akustiske impedans. Til venstre: kvartbølgelag ($\ell_k = \lambda_k/4$), til høyre: 3/4-bølgelag ($\ell_k = 3\lambda_k/4$).

2.3.1 Spesifikk akustisk impedans

Ved hjelp av ligning (1.37) og materialparametere fra tabell 2.3, ble den forventede admittansen beregnet for en piezokeramisk skive med koblingslag ved lydutstråling i vann.

Når koblingslagets akustiske impedans øker relativt til den i vannet, vokser systemets båndbredde. Resonanstoppen deler seg gradvis i to smalere topper, en på hver side av elementets resonansfrekvens. Dette går frem av figur 2.10, som viser simulerte konduktanser fra Mason-modellen.

En passende akustisk impedans for kvartbølgelaget ble funnet rundt 4,5 MRayl (60% kobber), som gav en frekvensrespons med bred og forholdsvis flat topp (figur 2.11). For 3/4-bølgelaget ble det funnet en tilsvarende forventet respons, men med noe smalere båndbredde, ved 3,4 MRayl. For koblingslag med tykkelse $5\lambda_k/4$ og større, hadde epoxymaterialet alene for stor akustisk impedans til å gi en god akustisk kobling.

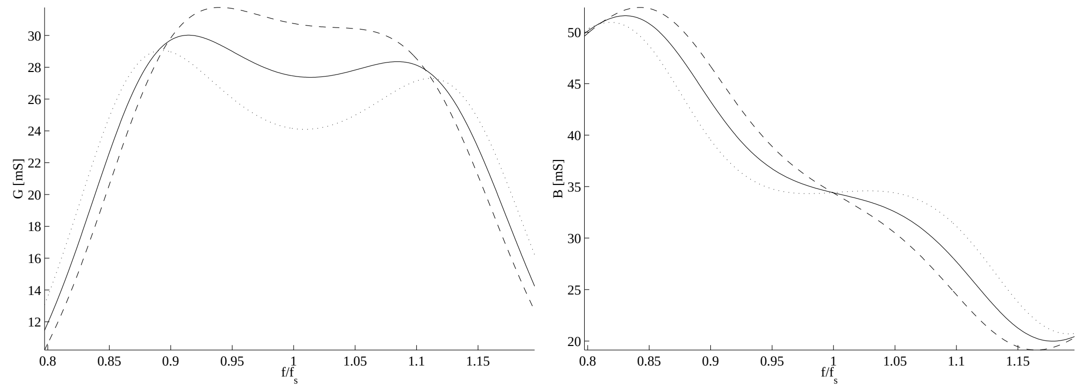
2.3.2 Valg av tykkelse

Det optimale blandingsforholdet for et kvartbølgelag med kobber ga en forholdsvis lav lydfart, og koblingslagets tykkelse ble beregnet til ca. 0,47 mm.

Det ble også brukt et trekvart-bølgelag. Impedansen ble i dette tilfellet noe lavere, og den ønskede tykkelsen ble satt til 1,68 mm. Figur 2.12 viser hvordan den beregnede konduktansen varierte med koblingslagets tykkelse ved to ulike spesifikke akustiske impedanser.

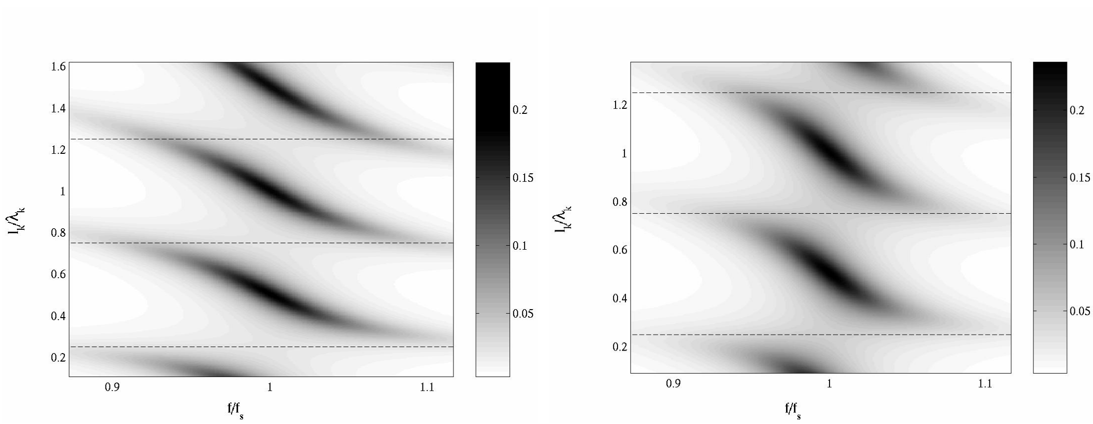
Når λ_k er bølgelengden i koblingslaget ved resonansfrekvensen, og tykkelsen

$$\ell_k = \frac{\lambda_k}{4} + n \cdot \frac{\lambda_k}{2}, \quad n = 0, 1, 2, \ldots,$$



Figur 2.11: Frekvensrespons for element med kvartbølgelag (konduktans t.v., susceptans t.h.) Stiplet kurve: $z_k = 4,3$ MRayl, heltrukken kurve: $z_k = 4,5$ MRayl, prikket kurve: $z_k = 4,8$ MRayl.

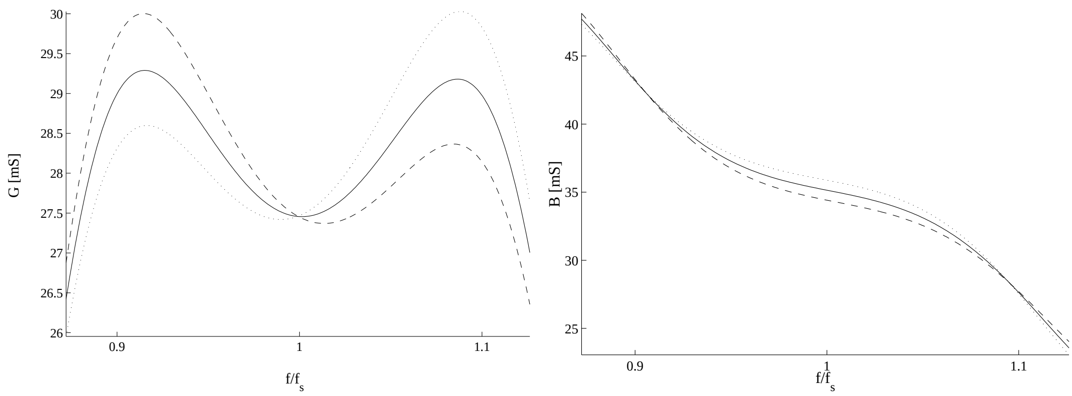
blir resonansen ifølge Mason-modellen asymmetrisk. Den venstre av de to toppene i konduktansen blir noe større enn den høyre. Dette kan det i teorien kompenseres for ved å øke frontlagets tykkelse som vist på figur 2.13. Den nødvendige tykkelsesjusteringen var på størrelse med ubestemtheten i elementets tykkelse (tabell 2.1), slik at en viss asymmetri i konduktansen ble forventet.



Figur 2.12: Konduktans i Siemens i elementet som funksjon av frekvens og koblingslagets tykkelse. Til venstre: $z_k = 4,5$ MRayl (60% Cu), til høyre: $z_k = 3,4$ MRayl (29% Cu). $\ell_k = \lambda_k/4$, $\ell_k = 3\lambda_k/4$ og $\ell_k = 5\lambda_k/4$ er avmerket med stiplede linjer.

2.3.3 Virkning på impulsresponsen

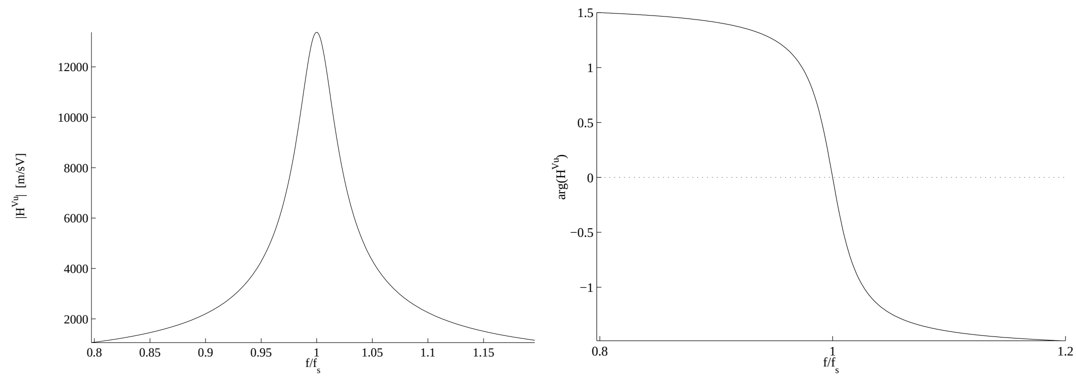
Den beregnede overføringsfunksjonen $H^{Vu}(f)$ fra spenning inn på et piezoelektrisk element uten koblingslag, til partikkelfart på grenseflaten mot vann, ble funnet



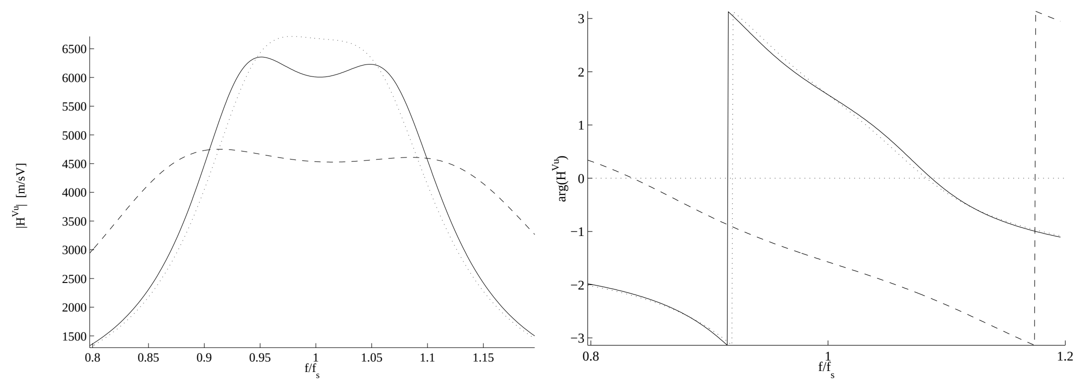
Figur 2.13: Frekvensrespons for elementet (konduktans t.v., susceptans t.h.), $\varphi_m = 0,60$. Stiplet kurve: $\ell_k = 0,2500\lambda_k$, heltrukken kurve: $\ell_k = 0,2516\lambda_k$, prikket kurve: $\ell_k = 0,2533\lambda_k$.

ved hjelp av Mason-modellen (figur 2.14). Overføringsfunksjonene som på samme måte ble funnet med tre ulike koblingslag, er vist i figur 2.15.

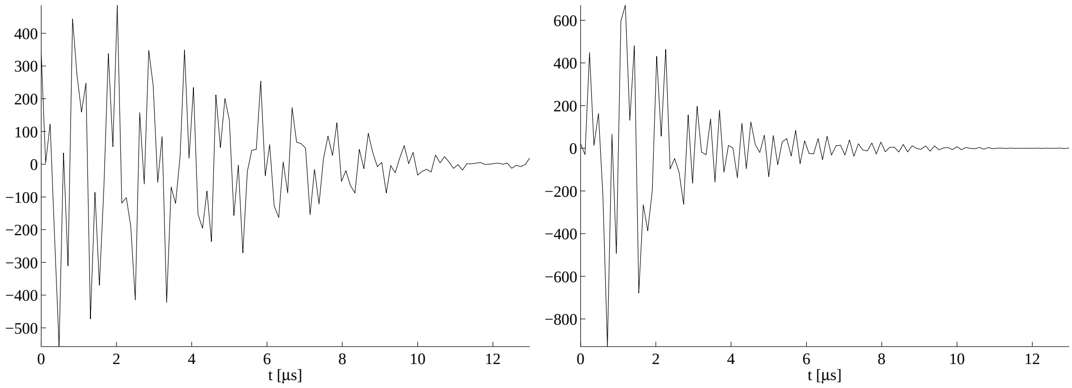
Ved å trunkere overføringsfunksjonene ved 4 MHz og Fouriertransformere dem ved hjelp av IFFT-rutinen i Matlab, ble det oppnådd impulsresponser som i figur 2.16 og 2.17. Kvartbølgelaget så ut til å gi den bredeste resonansen og den korteste impulsresponsen.



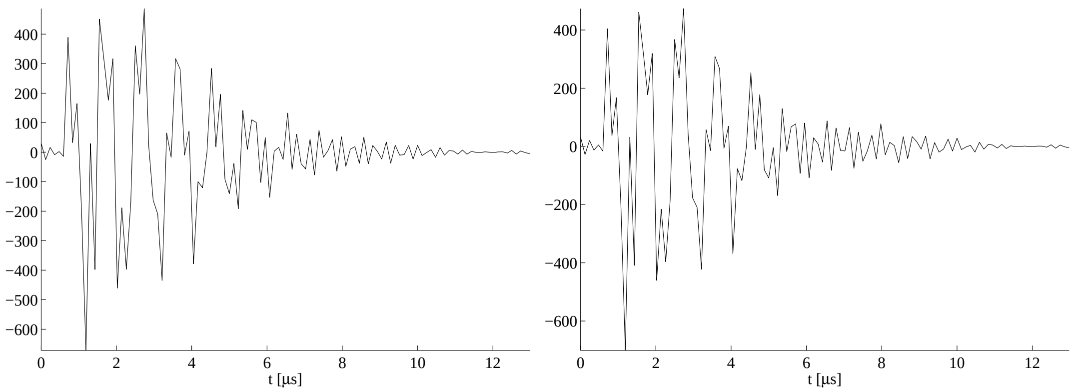
Figur 2.14: Overføringsfunksjoner fra spenning inn på det piezoelektriske elementet til partikkelfart på kildens overflate. Absoluttverdi og fase, uten frontlag.



Figur 2.15: Overføringsfunksjoner fra spenning inn på det piezoelektriske elementet til partikkelfart på kildens overflate (utenpå koblingslaget). Heltrukken kurve: $\ell_k = 3\lambda_k/4$, 29% Cu. Stiplet kurve: $\ell_k = 1\lambda_k/4$, 60% Cu. Prikket kurve: $\ell_k = 3\lambda_k/4$, ren Araldite.



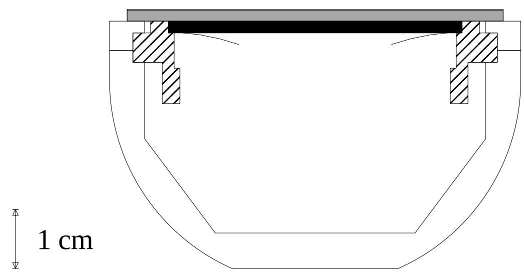
Figur 2.16: Beregnet impulsrespons for kilde som stråler i vann.
Til venstre: Uten koblinglag, til høyre: Koblingslag med tykkelse $\lambda/4$ og 60% Cu (4,5 MRayl)



Figur 2.17: Beregnet impulsrespons for kilde som stråler i vann. Til venstre: Koblingslag med tykkelse $3\lambda/4$, ren Araldite, til høyre: $3\lambda/4$, 29% Cu (3,4 MRayl)

2.4 Sammensetning av transduseren

Lydkilden ble laget av messing, med en plastring (POM) for oppheng og elektrisk isolasjon av det piezokeramiske elementet. Bak elementet ble det laget rom til en eventuell enhet for elektrisk impedanstilpasning. De viktigste delene av transduseren er skissert på figur 2.18.



Figur 2.18: Snitt gjennom transduseren. Viktige deler: Koblingslag, piezoelektrisk element (svart), messingshim for elektrisk tilkobling, plastring som holder elementet (skravert), ytre messingring, halvkuleformet bakre del av messing.

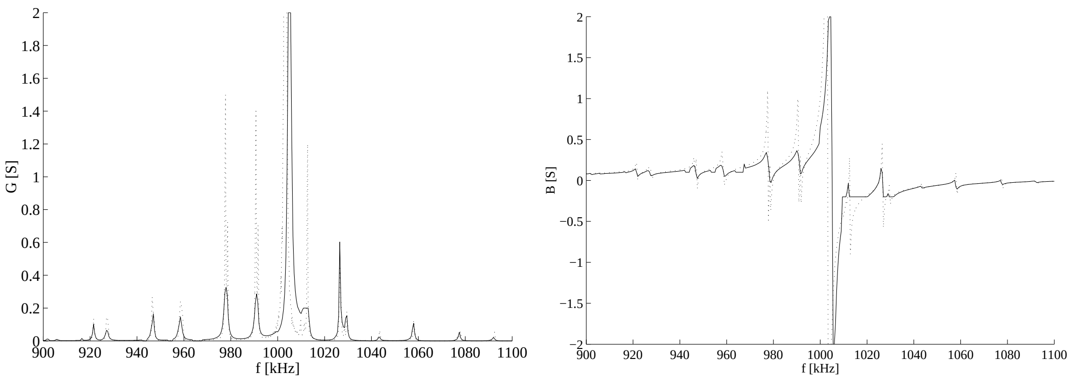
2.4.1 Montering av elementet

Sideflaten på den piezokeramiske skiven ble limt til plastringen ved hjelp av Sikaflex. Mellom elementets bakside og plastringen ble det lagt en ring av 0,25 mm messingfolie ("shim") for elektrisk tilkobling (figur 2.19). I tidligere transduserkonstruksjoner med luftbacking har det av og til vært loddet en tynn leder til baksiden av skiven, som har vært utsatt for varmeutvikling. Dette problemet var det viktig å unngå på grunn av etanoldampen og den eventuelle brannfaren i SOBER-tanken. Som en ekstra sikkerhet ble det laget et hull på baksiden av transduseren for fylling av nitrogen, slik at en eksplosiv gassblanding i rommet bak elementet kunne forebygges.

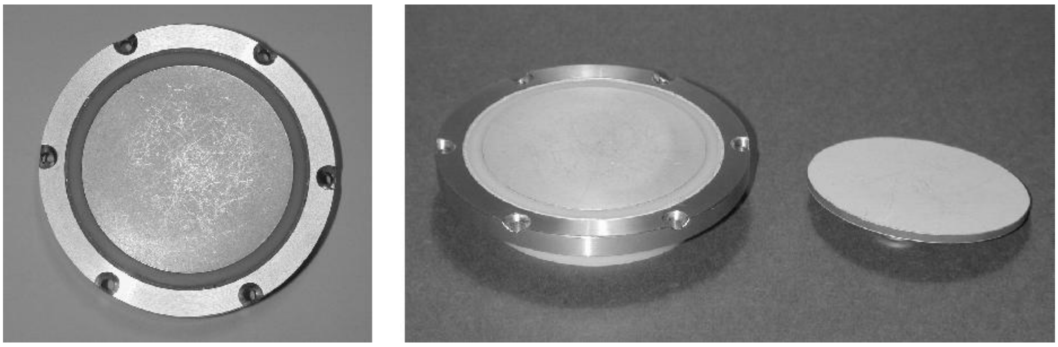
Sikaflex ble også benyttet mellom plast- og messingringene i forkant av transduseren. Etter montering av elementet (figur 2.21), endret admittansen seg som vist i figur 2.20.

For de løse skivene ble ikke impedansanalysatoren nulljustert ved åpen og lukket krets. Dette ga måleverdier som var større enn analysatorens nominelle maksimalverdi på 2 S. Begrensningen på måleområdet er årsaken til at konduktansen i tykkelsesmoden blir "kuttet" i figur 2.20, der nullkalibrering ble benyttet. I dette siste tilfellet, og for impedansmålingene i resten av denne oppgaven, ble målingen foretatt i to omganger, da frekvensene under og over 1 MHz tilhører hvert sitt gyldighetsområde for kalibrering.

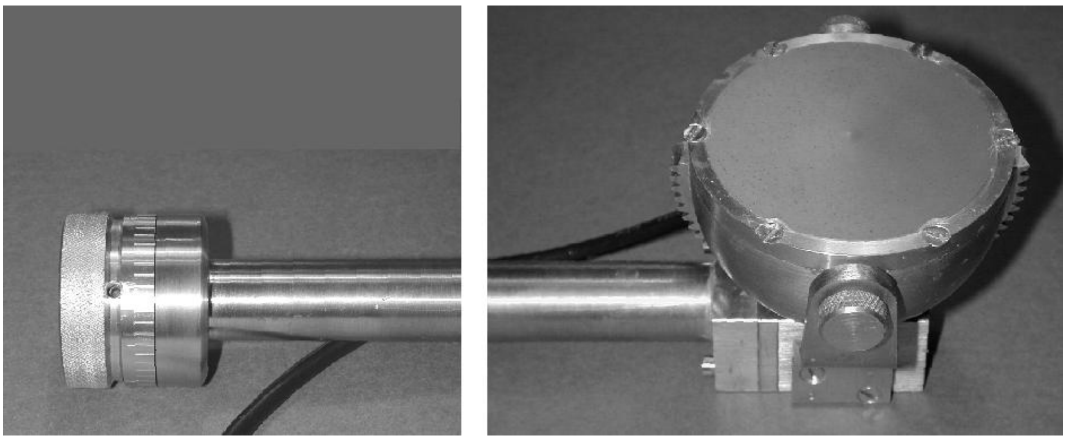
Figur 2.19: Elektrisk tilkobling til baksiden av elementet.



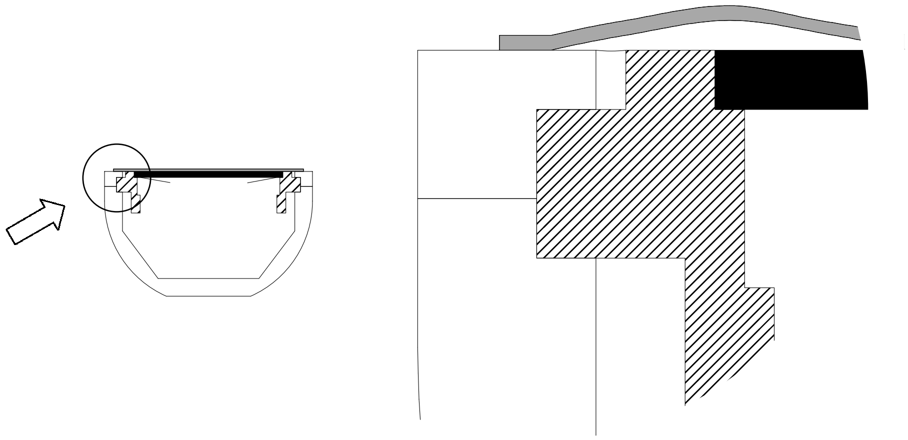
Figur 2.20: Konduktans og susceptans. Heltrukken kurve: Element 5D montert i plastring, stiplet: Element 5D før montering.



Figur 2.21: Til venstre og i midten: Fronten til transduseren før påliming av bladgull og støping av koblingslag, til høyre en av de piezokeramiske skivene.



Figur 2.22: Den ferdige lydkilden med kvartbølgelag.



Figur 2.23: Kvartbølgelaget løsnet fra transduseren der det var lagt bladgull.

2.4.2 Mekanisk koblingslag

Det ble valgt å støpe koblingslag av Araldite og kobberpulver direkte på kildens forside. Tykkelsen ble justert ved hjelp av en dreiebenk etter at støpemassen var stivnet. Plastringen ble brukt som referanse under dreiingen, og det var monteringen av elementet i denne som ga det viktigste bidraget til ubestemtheten i frontlagets tykkelse.

Det første frontlaget som ble støpt på kilden, var et kvartbølgelag med 60% kobber. Før støpingen ble det lagt bladgull i en sirkel, fra messingringen og ca. 5 mm inn på elementets fremre elektrode. Bladgullet ble limt fast ved hjelp av strømledende maling (Grace Eccocoat CC2). Etter en kort periode i drift, trengte det inn vann mellom bladgullet og sølvmalingen. Frontlaget mistet dermed den mekaniske kontakten med transduseren rundt hele elementets ytterkant (figur 2.23).

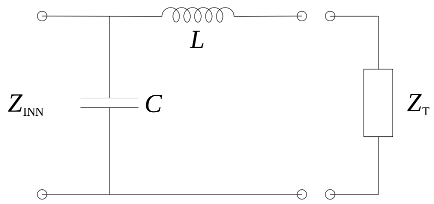
Kvartbølgelaget ble erstattet av et 3/4 bølgelengde tykt koblingslag av Araldite med 23% kobber. Istedenfor å legge bladgull langs hele ytterkanten av det piezokeramiske elementet, ble det lagt seks halvcentimeter-brede striper i radiell retning, fra messingringen og 5 mm inn på elementet. En skive av transparent-plast ble deretter lagt slik at den dekket plast- og messingringene, men ikke det piezokeramiske elementet. Dette ble gjort for å hindre skjærkrefter fra de passive delene av transduseren i å virke på frontlaget. For å hindre vanninntrengning mellom messingringen og den nye plastskiven, ble det lagt myk silikon rundt kanten av koblingslaget.

2.4.3 Elektrisk impedanstilpasning

Den utstrålte effekten fra kvartbølgelaget ble økt ved hjelp av et forenklet $\pi-$filter. Siden elementet representerte en kapasitiv last i det interessante frekvensområdet, ble kondensatoren på transdusersiden av filteret tatt bort (figur 2.24).

Tabell 2.6: Parameterne som ble brukt ved tilpasning av Mason-modellen til målte admittanser.

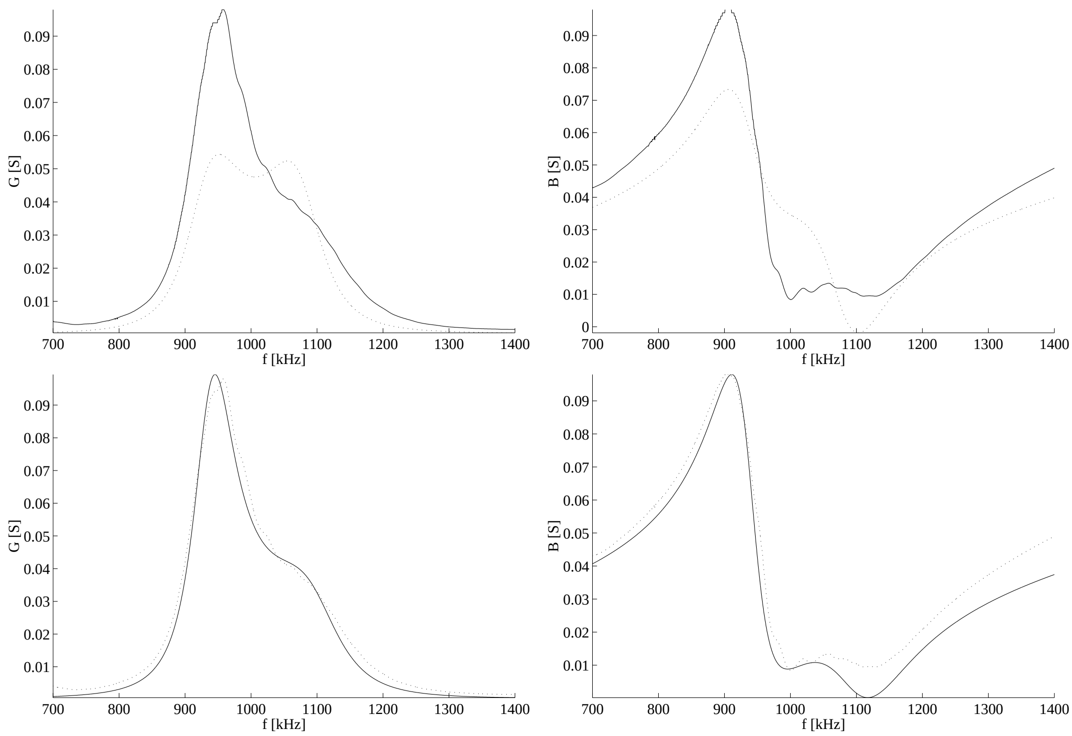
		Kvartbølgelag		3/4-bølgelag	
		Opprinnelig	Tilpasset	Opprinnelig	Tilpasset
c_{33}^D	[10^{11} N/m^2]	1,57	1,6	1,57	1,6
ε_{33}^S	[10^{-9} C/Vm]	6,7	6,4	6,7	5,8
e_{33}	[C/m^2]	14,9	16	14,9	15,5
ℓ_k	[mm]	0,466	0,385	1,676	1,615
z_k	[MRayl]	4,5	5,1	3,3	3,3
φ_m		0,60	0,68	0,23	0,23



Figur 2.24: Modifisert $\pi-$filter.

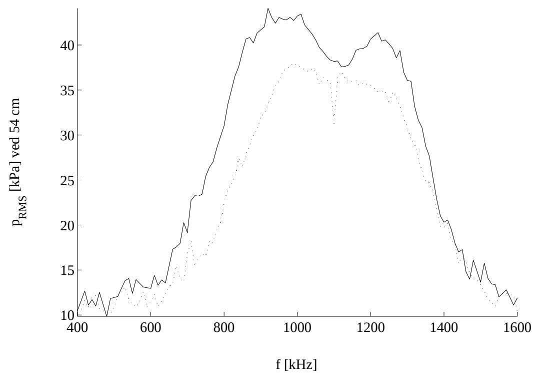
Transduserens admittans ved lydutstråling i vann ble målt ved hjelp av impedansanalysatoren, og parameterne i Mason-modellen ble endret for å få et simuleringsresultat som passet bedre med det målte (tabell 2.6). Figur 2.25 viser tilpasningen som ble gjort for trekvart-bølgelaget. Den tilpassede simuleringen ble brukt til å estimere de verdiene for L og C som ga en minst mulig susceptans og en resistans nærmest mulig 50 Ω ved transduserens serieresonans.

Tilpasningskretsen ble finjustert ved å se eksperimentelt på hvilke komponentverdier som ga den største utstrålte lydamplituden. For kvartbølgelaget ble induktansen satt til 6,6 μH, og det ble brukt en kondensator på 4,7 nF. I tilfellet med trekvart-bølgelag ble den beste frekvensresponsen oppnådd uten en elektrisk koblingsenhet. Driftsfrekvensen ble i det siste tilfellet økt fra en til 1,1 MHz for å få høyest mulig effekt ut av systemet og samtidig beholde en tilstrekkelig kort transientrespons.



Figur 2.25: Øverst: Admittans for transduseren med 3/4-bølgelag, målt (—) og forventet (···). Nederst: Samme måleverdier, men med endrede parametere i Mason-simuleringen (tabell 2.6).

Figur 2.26 viser trykkamplituder på aksen 54 cm fra lydkilden med kvartbølgelag, med og uten den elektriske impedanstilpasningsenheten som ble brukt ved målinger på lydfeltet.



Figur 2.26: Målt lydtrykk på aksen 54 cm fra kilden med kvartbølgelag. Med (—) og uten (· · ·) elektrisk koblingsenhet.

2.4.4 Posisjonering

Kilden ble montert med mulighet for vertikal forskyvning og rotasjon. Dreiningen ble gjort ved hjelp av et snekkehjul på transduserens bakside (figur 2.22). Inne i holderen på baksiden av transduseren, ble det montert en snekkeskrue som kunne roteres ved hjelp av en aksling i holderens 55 cm lange rør. På denne måten kunne man fra toppen av røret rotere transduseren vertikalt med en toleranse på en tidels grad.

2.5 Måling av lydfelt

2.5.1 Instrumentoppstilling

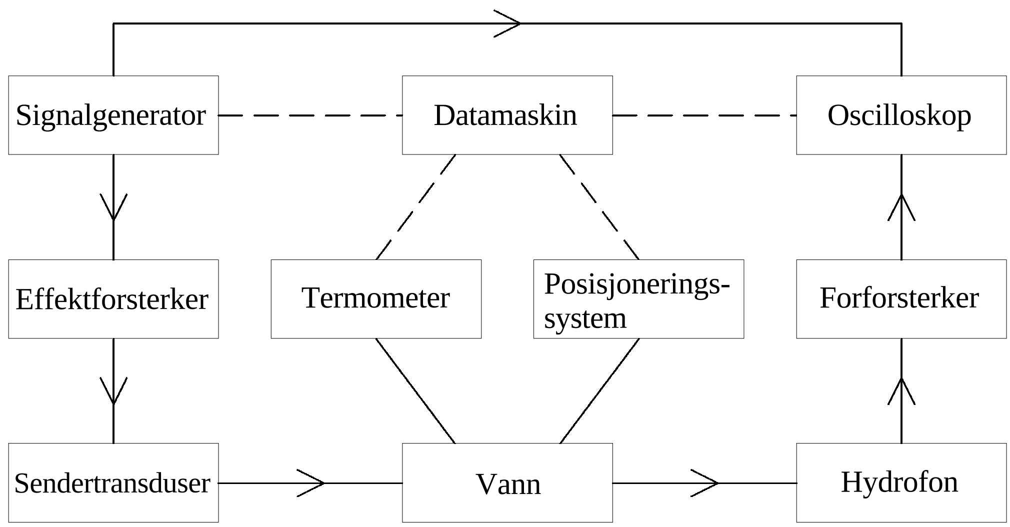
Strålingsfeltet ved lav amplitude ble undersøkt for begge de to koblingslagene som ble brukt i oppgaven, og med 3/4-bølgelaget ble det også målt på ikke-lineære lydfelt. Det ble gjort målinger i to tanker. Måleoppsettet i tank 1 er beskrevet i detalj av Kippersund [37]. I tank 2 ble det brukt en annen hydrofon, men måleoppstillingen ellers var den samme. Den siste tanken var bygget spesielt for SOBER-eksperimentet. Tabell 2.7 og figur 2.27 gir en oversikt over de benyttede instrumentene.

Tabell 2.7: Instrumenter som ble brukt ved undersøkelse av lydfeltet fra transduseren.

Felles	
Datamaskin	Compaq PC med Matlab 4.2 og GPIB-kort
Funksjonsgenerator	HP 33120A
Effektforsterker	ENI 240L, 50 dB
Sendertransduser	Laget i denne oppgaven
Oscilloskop	LeCroy 9350
Termometer	ASL F250
Kabler	RG-58, spesifikk impedans 50 Ω
Oppsett 1	
Vanntank	Glass, lengde 198 cm, bredde 48 cm, høyde 102 cm
Hydrofon	GEC-Marconi Y-34-3598 PVDF membranhydrofon
Posisjoneringssystem	Coradi (Styringssystem av Hobæk og Lien)
Forforsterker 40/60 dB	Panametrics 5660B
Oppsett 2	
Vanntank	Glass, lengde 270 cm, bredde 50 cm, høyde 70 cm
Hydrofon	Precision Acoustics PVDF nålhydrofon
Posisjoneringssystem	Coradi (Hobæk, Lien og Fardal)
Likespenningskilde	Mascot 719

Måleoppsettet besto av en vanntank, en hydrofon montert i et posisjoneringssystem, lydkilden, senderelektronikk, mottagerelektronikk, et termometer og en PC. Datamaskinen styrte instrumentene og posisjoneringssystemet gjennom GPIB. Måledataene fra oscilloskopet og termometeret ble på samme måte hentet inn og lagret. Programmeringen av måleseriene ble gjort i Matlab 4.2. Programmer av Hobæk, Berge og Kippersund ble brukt i forbindelse med styring av eksperimentet og til datainnsamling.

For å unngå at refleksjoner fra vanntankenes vegger skulle påvirke det målte

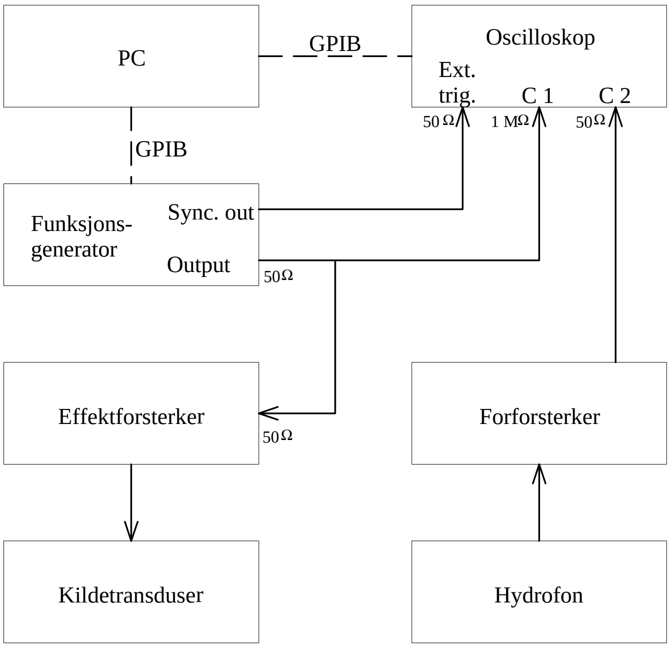


Figur 2.27: Skisse av signalgangen mellom instrumentene i tabell 2.7. Stiplede linjer er GPIB-forbindelser. Pilene viser signalenes gangretning gjennom koaksialkabler og i vannet.

signalet, ble det brukt sinus-pulser ("bursts") med varighet på 20–30 perioder. Signalet fra funksjonsgeneratoren kunne justeres i amplitude fra 50 $\mathrm{mV_{P-P}}$ til 10 $\mathrm{V_{P-P}}$. Det ble videre forsterket av en ENI 240L effektforsterker.

Kildetransduseren ble plassert midt på vanntankens ene kortside og rettet inn med tanke på å gjøre lydaksen mest mulig parallell med posisjoneringssystemets z-akse. Posisjoneringssystemet som flyttet hydrofonen rundt i tank 1, var utstyrt med tre stegmotorer som flyttet hydrofonen i x-, y- og z-retning. z-aksen ble valgt i tankens lengderetning, x-aksen horisontalt på tvers av lydaksen, og y-aksen i vertikal retning. Stegmotorene på y- og z-aksene hadde i begge tankene 1:10-girutvekslinger som økte nøyaktigheten. Avstanden hydrofonen ble forflyttet for hvert steg er satt opp i tabell 2.8. I oppsett 1 kunne hydrofonen plasseres hvor som helst i tanken, mens bevegelsen i tank 2 var begrenset i z-retning til mellom 84 cm og 250 cm fra kilden. Tank 2 manglet også stegmotor for forflytning i x-retningen.

I den første tanken ble det brukt en bilaminær PVDF-membranhydrofon. Den besto av to sirkulære, 25 μm tykke membraner, med et aktivt område i sentrum med diameter 0,5 mm. I denne oppgaven ble det brukt kalibreringsdata fra fabrikanten (29.03.1995) [38], der det er oppgitt mottakerfølsomhet M_C, resistans R og reaktans X for hver hele MHz fra 1 til 20.



Figur 2.28: Koblingsskjema for de viktigste instrumentene i måleoppsettet. De stiplede linjene representerer GPIB-forbindelser, mens de heltrukne viser koblinger gjort med koaksialkabler. Med unntak av forbindelsen mellom membranhydrofonen og forforsterkeren, er alle koaksialkablene av typen RG-58.

Tabell 2.8: Forflytning pr. steg for de to posisjoneringssystemene.

	x-retning [mm/steg]	y-retning [mm/steg]	z-retning [mm/steg]
Oppsett 1	0,098	1/320	1/24
Oppsett 2	–	0,003125	1/24

Når membranhydrofonens utgang belastes med en impedans $Z_{\mathrm{EL}} = R_{\mathrm{EL}} + iX_{\mathrm{EL}}$, blir den resulterende mottakerfølsomheten [38]

$$M_{\mathrm{L}} = M_{\mathrm{C}}\sqrt{\frac{|Z_{\mathrm{EL}}|^2}{|Z_{\mathrm{EL}} + Z|^2}}.$$

Z er her impedansen over hydrofonens utgang, og M_{C} er frittfelts-mottakerfølsomheten for åpen krets. I kalibreringstabellen er denne oppgitt i enden av hydrofonens 71 cm lange kabel.

Hvis hydrofonens og lastens impedanser ses på som rent kapasitive, kan uttrykket over forenkles til

$$M_{\mathrm{L}} \approx M_{\mathrm{C}}\frac{C}{C + C_{\mathrm{EL}}}, \tag{2.1}$$

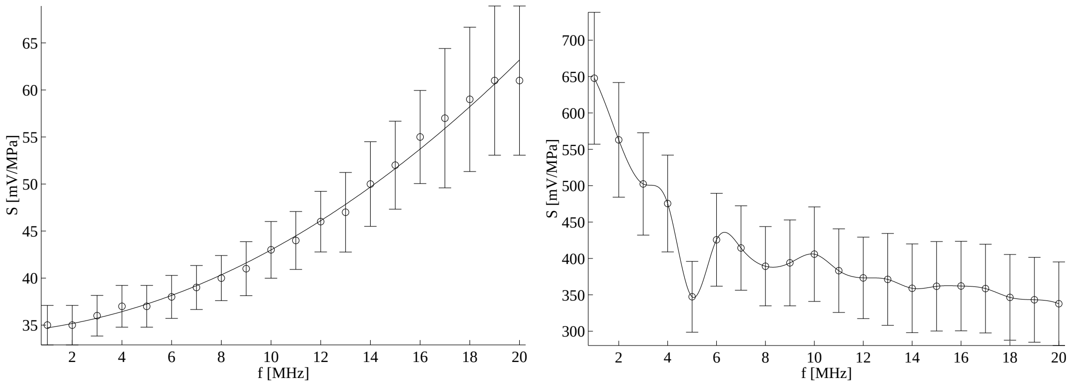
der $C \approx -1/(\omega X)$ og C_{EL} er kapasitansene i henholdsvis hydrofonen og kretsen som belaster den.

Ligning (2.1) ble brukt for å kompensere for lasten fra forforsterkeren der denne ble brukt, og ellers fra en signalkabel og oscilloskopet. På hydrofonens mottakerfølsomhet og reaktans ble det brukt henholdsvis polynomapproksimasjon og spline-interpolasjon for å oppnå interpolerte verdier (figur 2.29 og 2.30).

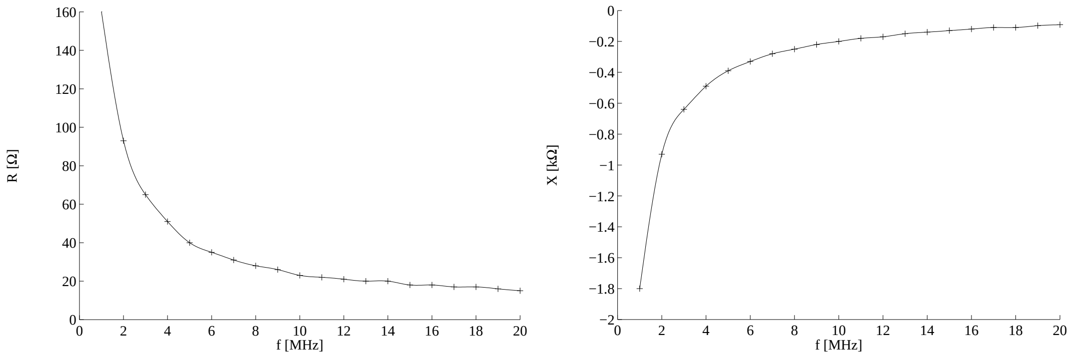
Forforsterkeren av typen Panametrics 5660B hadde liten båndbredde [37], og ble derfor bare brukt ved målinger på lydfelt med lav amplitude. Forsterkningen ved 1005 kHz og 1100 kHz ble målt ved hjelp av LeCroy-oscilloskopet. Inngangssignalet ble laget ved hjelp av signalgeneratoren og en dempeenhet med utgangsimpedans 50 Ω. Målingene ble gjort med faste oscilloskop-innstillinger, 2 mV/DIV for inngangsspenningen og 10 mV/DIV på forsterkerens utgang (50 Ω). Resultatene er vist i figur 2.31.

Hydrofonen som ble brukt i oppsett 2, var en Precision Acoustics nålhydrofon. Det aktive området i enden av nålen var av 28 μm tykk PVDF og hadde diameter 0,5 mm. En 8 dB forforsterker var integrert i hydrofonen, og ble drevet ved hjelp av en likespenningskilde. Utgangsimpedansen fra forforsterkeren var 50 Ω. Kalibreringsdata fra produsenten var oppgitt i området 1 MHz–20 MHz [39]. Følsomheten er plottet i figur 2.29.

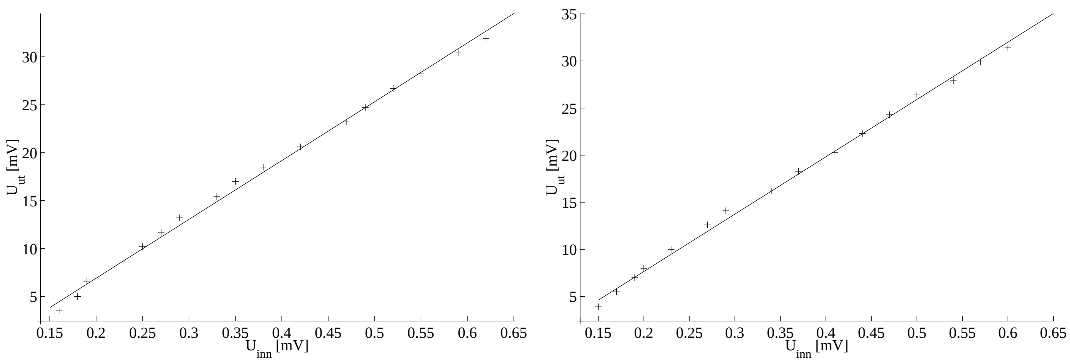
Til datainnsamling ble det brukt et LeCroy 9350 digitalt oscilloskop med 500 MHz samplingfrekvens. Oscilloskopet mottok et triggesignal fra funksjonsgeneratoren, som sammen med lydfarten og posisjoneringssystemets z-koordinat



Figur 2.29: Oppgitt mottakerfølsomhet for hydrofonene med ubestemtheter og tilpassede kurver. Til venstre: GEC Marconi membranhydrofon, åpen krets. Til høyre: Precision Acoustics nålhydrofon, 50 Ω.



Figur 2.30: Oppgitt impedans for membranhydrofonen (resistans t.v. og reaktans t.h.).



Figur 2.31: Til venstre: Panametrics-forforsterkerens amplituderespons ved 1005 kHz, målte punkter og tilpasset kurve (minste kvadraters metode). Kryssenes utstrekning viser ubestemtheten ved avlesning av oscilloskopet. Til høyre: Tilsvarende ved 1100 kHz.

ble brukt til å finne det ønskede utsnittet av lydpulsen. Lengden av de lagrede bølgeformene var 502 punkter, og det ble brukt 8-bits oppløsning. Oscilloskopets måleområde ble for hver enkelt måling justert til det minste som omsluttet hele bølgen.

Et digitalt termometer ble brukt for å overvåke temperaturen i vannet under noen av målingene.

Figur 2.28 viser hvordan de viktigste instrumentene var koblet sammen. Forforsterkeren som er tegnet inn mellom hydrofonen og oscilloskopet, ble bare brukt sammen med membranhydrofonen.

2.5.2 Strålingsfelt ved lav amplitude

Det ble gjort en rekke målinger i transduserens strålingsfelt, både langs lydaksen og i plan normalt på den. I tank 2 var det ikke mulig å bevege hydrofonen i x-retning. Målingene som ble gjort på tvers av lydaksen i denne tanken, ble tatt langs vertikale linjer.

Ved bruk av kvartbølgelaget ble signalet inn på effektforsterkeren satt til 100 $\mathrm{mV_{P-P}}$. 3/4-bølgelaget ga en noe større kildefølsomhet, og siden det ble observert en viss forvrengning av de målte lydbølgene, ble amplituden i det siste tilfellet redusert til 80 $\mathrm{mV_{P-P}}$. Den elektriske impedanstilpasnings-enheten ble brukt ved alle målinger på feltet fra kvartbølgelaget. Inngangssignalets frekvens var 1005 kHz for kvartbølgelaget og 1100 kHz for trekvart-bølgelaget.

2.5.3 Ikkelineære lydfelt

Trykket langs lydaksen ble målt ved ulike amplituder. Feltet på tvers av aksen ble også undersøkt ved utvalgte amplituder og avstander fra kilden. Det ble brukt sinus-pulser med varighet på 25 perioder i tank 1, og 30 perioder i tank 2.

Tabell 2.9: Inngangsamplituder for effektforsterkeren ved undersøkelse av ikkelineære lydfelt. Antall pulser det ble midlet over i hvert tilfelle.

Målesett nr.	Amplitude [$\mathrm{mV_{P-P}}$]	Pulser
1	200	10
2	300	10
3	400	5
4	600	10

I hver posisjon ble et utsnitt på 5 μs av lydbølgen lagret. Hvert av de 502 punktene ble midlet over et antall pulser (tabell 2.9). Ved beregning av bølgens Fouriertransform ble signalet kortet ned til 454 punkter, for å oppnå kontinuitet i den periodiske utvidelsen. Dette reduserer frekvenslekkasjen ved diskret Fouriertransformasjon (DFT) [40],[41].

DFT-algoritmen i Matlab ble brukt for å dekomponere de ikkelineære bølgeformene. Transformasjonen er definert som [42]

$$X_k = \sum_{n=1}^{N} x_n e^{-\frac{j\,2\pi(k-1)(n-1)}{N}}, \quad k = 1, 2, \ldots, N, \tag{2.2}$$

der N er signalets lengde. Det diskrete signalet x_n, tatt ved tidspunktene $t_n = t_0 + n\,\Delta t$, kan uttrykkes ved den reelle Fourierrekken

$$x_n = a_0 + \sum_{k=1}^{N/2} \left(a_k \cos \frac{2\pi k t_n}{N\,\Delta t} + b_k \sin \frac{2\pi k t_n}{N\,\Delta t} \right), \tag{2.3}$$

der koeffisientene er gitt ved Fouriertransformen (2.2) som

$$a_0 = \frac{X_1}{N}, \quad a_k = \frac{2\ \mathrm{Re}\{X_{k+1}\}}{N}, \quad b_k = -\frac{2\ \mathrm{Im}\{X_{k+1}\}}{N}.$$

Ved hjelp av relasjonen $\sin(g+h) = \sin g\,\cos h + \sin h\,\cos g$, kan ligning (2.3) skrives om til

$$x_n = c_0 + \sum_{k=1}^{N/2} c_k \sin(\frac{2\pi k t_n}{N\,\Delta t} + \gamma_k) = c_0 + \sum_{k=1}^{N/2} c_k \sin(\omega_k t_n + \gamma_k), \quad n = 1, 2, \ldots, N,$$

der amplituden c_k og fasen γ_k for hver frekvenskomponent er gitt ved

$$c_0 = a_0 = \frac{X_1}{N}, \quad c_k = \frac{2}{N}|X_{k+1}|, \quad \gamma_k = -\tan^{-1}\frac{\mathrm{Re}\{X_{k+1}\}}{\mathrm{Im}\{X_{k+1}\}}, \quad k = 1, 2, \ldots, N/2,.$$

$\omega_k = 2\pi\frac{k}{N\,\Delta t}$ er frekvensen som tilsvarer den k'te DFT-komponenten.

Hver harmoniske frekvenskomponent ble korrigert for hydrofonens følsomhet. Ved bruk av membranhydrofonen ble det tatt hensyn til den kapasitive lasten fra signalkabelen og oscilloskopet [38].

2.6 Endelig differanse-simulering av lydfelt

2.6.1 Den transformerte bølgeligningen

Det ble forsøkt å oppnå samsvar mellom Bergenkoden og de målte lydfeltene fra transduseren. Ligningen som passet best for problemet, var den transformerte bølgeligningen for plane og svakt fokuserende kilder (TBE, avsnitt 1.4.1). Ligningen ble løst med en plan stempelkilde som startbetingelse. Det ble ikke gjort viktige endringer i koden, som er listet i appendiks A (`tbe_plan_fok.f`).

Alternative kildebetingelser

Kildebetingelsen for den transformerte bølgeligningen spesifiseres i et plan, og en svakt kurvet kilde kan representeres ved en ekvivalent plan kildebetingelse. Ved å se bort fra ikkelineære effekter mellom kilden og planet der startbetingelsen blir beregnet, kan man utlede den Fresnel-approksimerte ekvivalente kildebetingelsen for en svakt kurvet kilde [43],

$$U(x) \sim u_s(x)\, e^{-ik\frac{x^2}{2d}}. \tag{2.4}$$

Her er U den ekvivalente plane kildebetingelsen, u_s partikkelhastigheten på kildens overflate, d kildens krumningsradius og k bølgetallet. Ved innsetting av den normaliserte avstanden fra symmetriaksen, $\xi = |\boldsymbol{\xi}| = |\mathbf{x}|/a$ der a er kilderadien, og den første ordens relasjonen $p_s = \rho_0 c_0 u_s$, kan man skrive (2.4) på formen

$$P(\xi) \sim p_s(\xi)\, e^{-ika^2\frac{\xi^2}{2d}}.$$

Et tilsvarende uttrykk finnes i J. N. Tjøtta og S. Tjøtta [24]. Bergenkoden ble kjørt med store d uten at man fikk resultatet til å nærme seg det for en plan kilde. Det ble forsøkt på ulike måter å kompensere for koordinattransformasjonen, uten at resultatet ble tilfredsstillende.

Det ble også forsøkt å løse KZK-ligningen med utgangspunkt i det målte feltet nær kilden. For 200, 300, 400 og 600 $\mathrm{mV_{P-P}}$ inngangsamplitude på effektforsterkeren, ble lydfeltet målt langs en linje vinkelrett på lydaksen. Avstanden fra linjen til kildeplanet var 29 mm. Måleresultatet ble normalisert og satt inn som startbetingelse i ligningen (`tbe_maalt.f`).

Verken den fokuserende/defokuserende eller den målte kildebetingelsen førte fram til et resultat som kunne konkurrere med beregningene for en plan stempelkilde. Det ble derfor ikke prioritert å behandle resultatene fra disse simuleringene videre.

2.6.2 MME og TEFB

Under forberedelsene av oppgaven ble det også eksperimentert med en versjon av Bergenkoden som tillater sterkere fokusering. Denne implementasjonen opererer med to ligninger; MME i den umiddelbare nærheten av kilden, og TEFB på resten av regneområdet. Overgangen mellom de to koordinatsystemene ble gjort ved hjelp av kubisk spline-interpolasjon i planet. Da NAG-programmene som opprinnelig ble brukt i interpolasjonen ikke lenger var tilgjengelige for instituttet, ble ACM-rutinen `itplbv.f` tatt i bruk som erstatning. På grunn av denne endringen er koden tatt med i appendiks A (`mbe_tefb.f`).

Kapittel 3

Resultat

3.1 Transduserens lineære strålingsfelt

3.1.1 Transientresponser

Rektangulære 10 perioders sinus-burster ble påtrykt kilden, og lydpulsene ble undersøkt for de to koblingslag-alternativene. Signalet inn på effektforsterkeren hadde peak-to-peak-amplitude 100 mV. Målingene ble gjort i vanntank 1 (figur 3.1), med membranhydrofon og forforsterker.

Med kvartbølgelag og elektrisk impedanstilpasning oppnådde transduseren fullt utslag etter omtrent tre perioder ved 1005 kHz. Bølgen så ut til å ha en oscillerende omhylningskurve, noe som kan tyde på en noe uren svingemode. Signalet døde ut ca. fire perioder etter at inngangssignalet hadde opphørt.

3/4-bølgelaget ga større stigetid enn kvartbølgelaget (fem perioder), mens svingningene døde ut litt raskere.

3.1.2 Feltet langs lydaksen

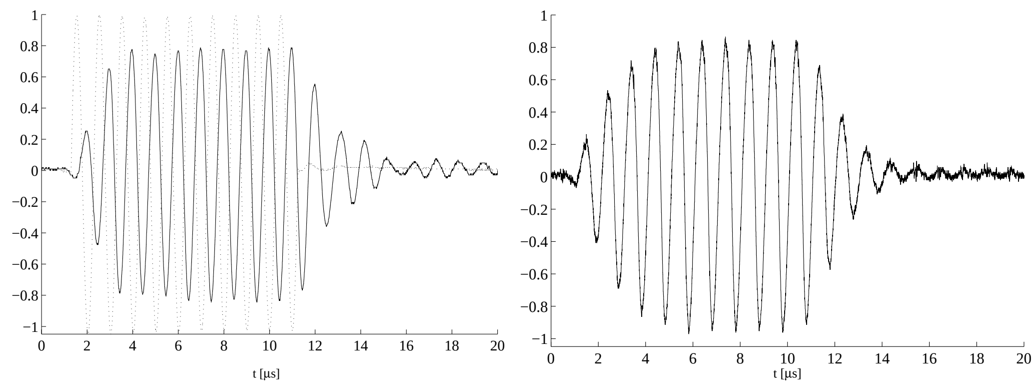
Figur 3.2 viser det målte RMS-lydtrykket på aksen med henholdsvis kvartbølge- og trekvart-bølgetykt koblingslag.

Kurven som har sitt siste maksimum lengst mot venstre, er for kvartbølgelaget ved 1005 kHz og amplitude 100 $\mathrm{mV_{P-P}}$ inn på effektforsterkeren.

Den andre kurven i figur 3.2 viser aksetrykket fra transduseren med 3/4-bølgelag. Kildefølsomheten ble høyere enn med kvartbølgelaget, selv om det her ikke ble gjort noen elektrisk impedanstilpasning. På grunn av begynnende ikkelineær oppførsel i lydfeltet, var inngangsamplituden til effektforsterkeren redusert til 80 mV. Frekvensen som ble brukt med 3/4−bølgelaget var 1100 kHz.

Ut fra uttrykket for avstanden til første maksimum i aksefeltet fra en plan sirkulær stempelkilde (1.64), kan det beregnes en effektiv kilderadius

$$a_e = \sqrt{z\lambda} = \sqrt{\frac{z\, c_0}{f}}.$$



Figur 3.1: Bølgeformer på aksen, 33 cm foran kilden. Inngangssignalet på effektforsterkeren var en 10 perioders sinusbølge ("burst") med rektangulær omhylningskurve. Til venstre: Med kvartbølgelag og elektrisk impedansomformer, 1005 kHz. Til høyre: Med 3/4-bølgelag, 1100 kHz.

a_e er radien til en plan, sirkulær stempelkilde som gir siste maksimum på lydaksen i samme avstand som den virkelige lydkilden. Tabell 3.1 viser de effektive kilderadiene som oppnås ved hjelp av måleverdiene i figur 3.2.

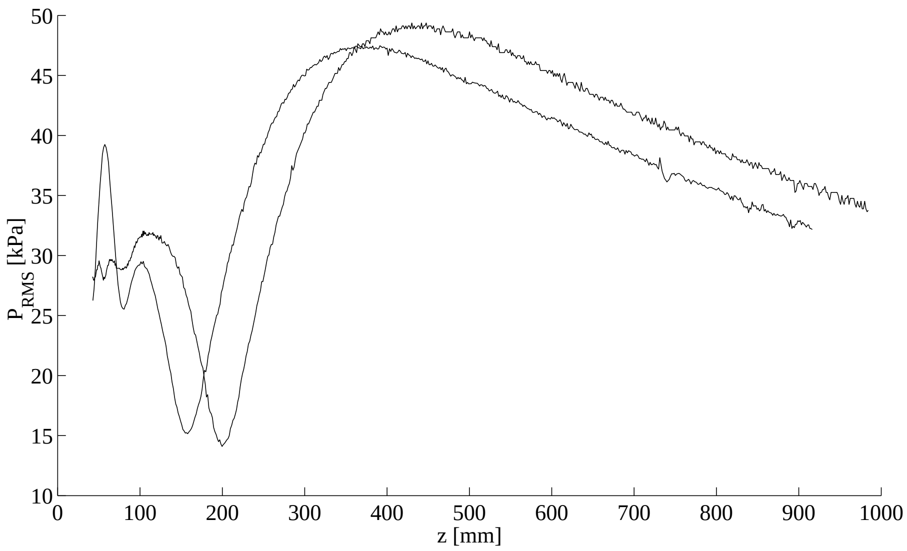
Tabell 3.1: Effektive kilderadier på grunnlag av det målte aksefeltet. Lydfarten c_0 er satt lik 1490 m/s.

Frontlag	z_1 [mm]	f [kHz]	a_e [mm]
$\lambda/4$	375±10	1005	23,6±0,3
$3\lambda/4$	435±10	1100	24,3±0,3

Ved hjelp av uttrykkene i ligning (1.67) og (1.68), kan man finne karakteristiske vinkler for fjernfeltet fra en plan stempelkilde med radius a_e. Verdiene er oppsummert i tabell 3.2. Hvis feltet fra trekvartbølgelaget ligner på det fra en plan, sirkulær stempelkilde med radius 24,3 cm, får hovedloben en bredde (minimum–minimum) på omkring 7 cm i to meters avstand fra kilden, og 10 cm ved tre meter.

Tabell 3.2: Karakteristiske vinkler for hovedloben til en plan stempelkilde med radius a_e.

	$\ell_k = \lambda/4$	$\ell_k = 3\lambda/4$
$\theta_{-3\,\mathrm{dB}}$	$0,93°\pm0,01°$	$0,82°\pm0,01°$
$\theta_{-6\,\mathrm{dB}}$	$1,28°\pm0,02°$	$1,13°\pm0,02°$
θ_1	$2,21°\pm0,03°$	$1,95°\pm0,02°$



Figur 3.2: Trykkamplituder langs lydaksen.

3.1.3 Feltet på tvers av aksen

Amplitude og fase ble målt i flere plan på tvers av lydaksen. Det ble brukt 47 punkter i hver retning, med en romlig oppløsning på 1,4 mm.

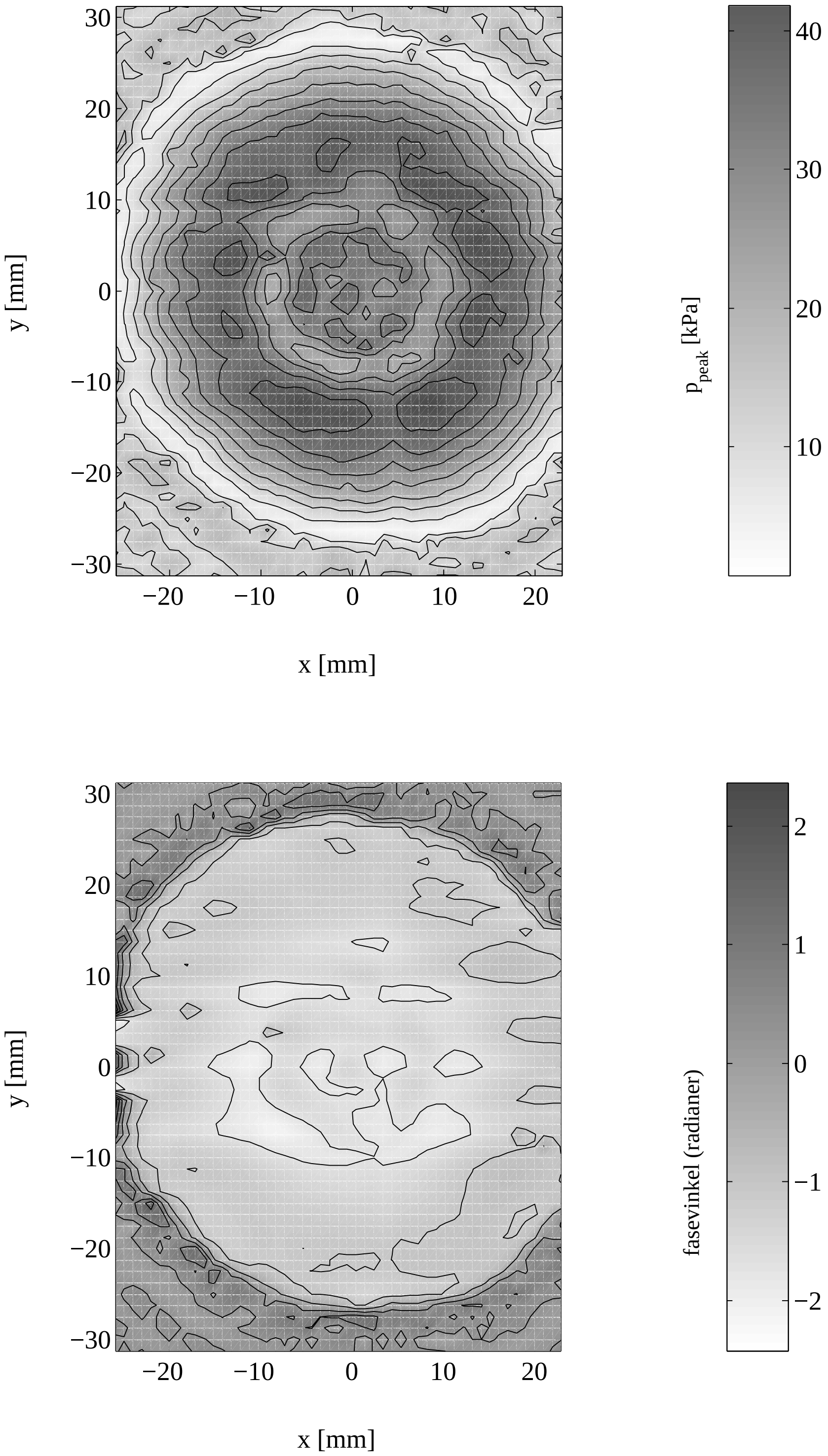
Amplituden og fasen som ble målt i planet 4,1 cm foran transduseren med kvartbølgelag og elektrisk tilpasningsenhet, er plottet på figur 3.3. Figur 3.4 viser amplitudefordelingen ved 54 cm.

Etter at 3/4-bølgelaget var montert, ble amplitudefordelingen noe mindre symmetrisk. Figur 3.6 viser amplitude og fase 4,1 cm fra kilden, og figur 3.7 viser det samme ved 75 cm.

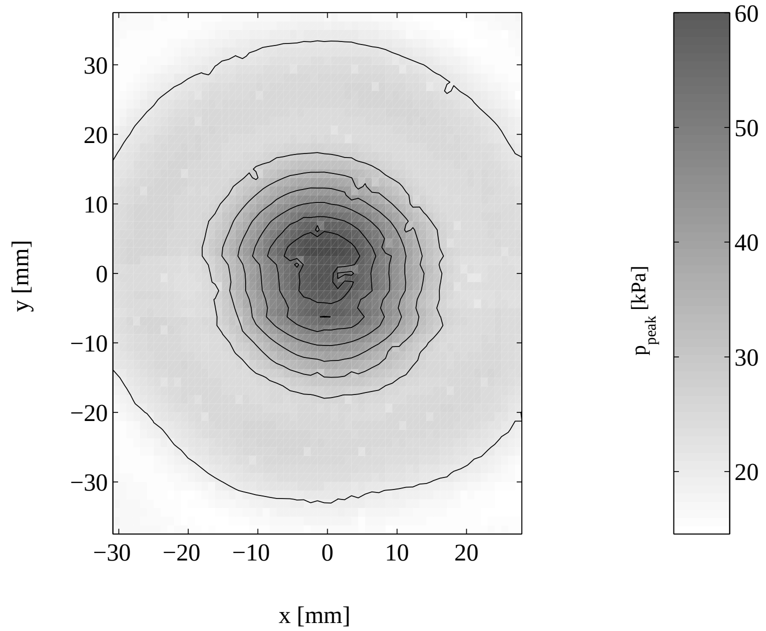
Målingene fra måleoppstilling 1 viser at fasen varierer periodisk med y-koordinaten med opptil en radian. Dette viste seg å skyldes en skjevhet i posisjoneringssystemet, som gjorde at hydrofonen beveget seg i z-retning når motoren for vertikal forskyvning ble kjørt. Ved 1100 kHz og lydfart 1483 m/s, tilsvarer en faseendring på 1 radian en forflytning på 0,2 mm.

Fasevariasjonen i vertikal retning ble målt på nytt i tank 2. Resultatene i avstand 149 cm og 214 cm fra kilden er vist i figur 3.8. Positiv retning på y-aksen er her nedover. Som det fremgår av figuren, ble den maksimale fasevinkelen målt et stykke nedenfor midten av strålingsfeltets hovedlobe. Dette tyder på at lydaksen og posisjoneringssystemets y-akse ikke sto helt vinkelrett på hverandre.

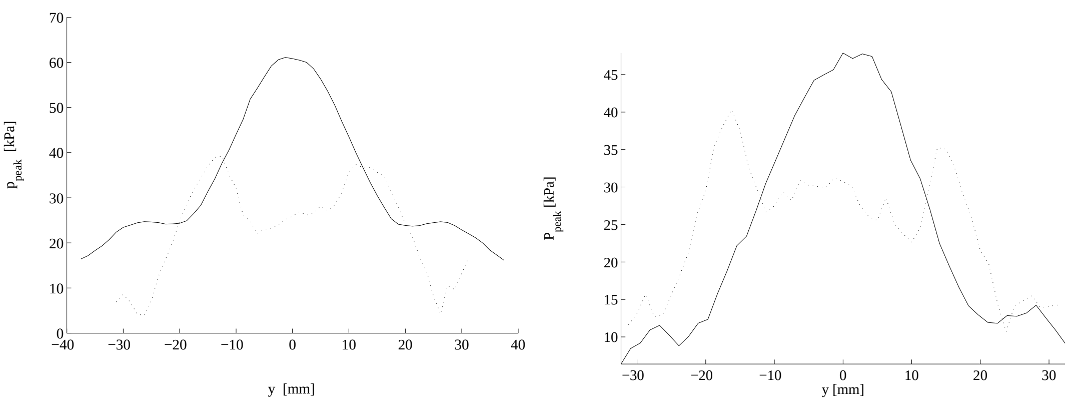
Man kan se små lokale maksima i fasen i nærheten av amplitudens minimalpunkter, der måleusikkerheten for fasen er størst. Ujevnhetenes avstand fra z-aksen øker med avstanden fra kilden på samme måte som for amplitudens minima. Det er derfor ikke sannsynlig at de er blitt forårsaket av feil i posisjoneringssystemet i måleoppsett 2.



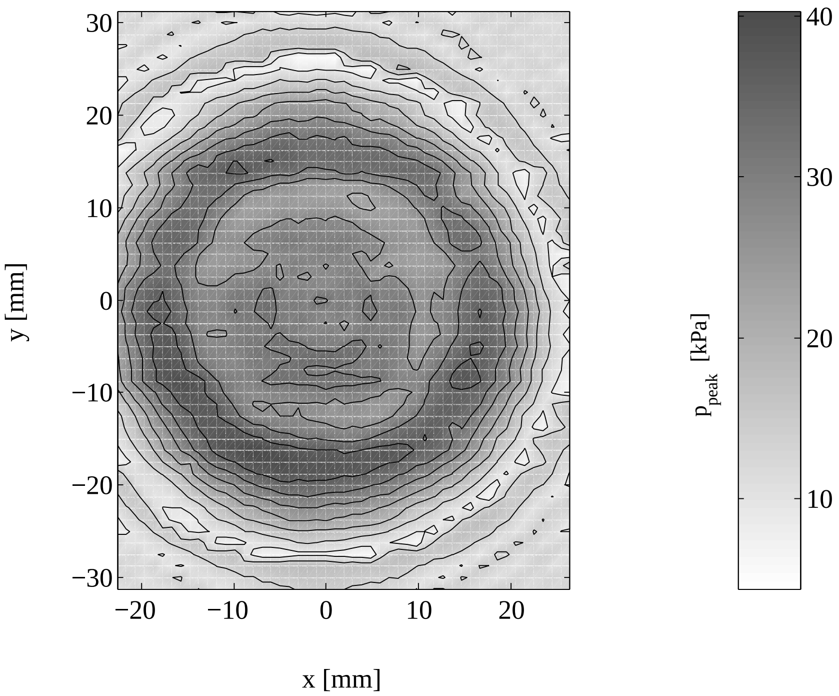
Figur 3.3: Kvartbølgelag, amplitude (øverst) og fase (nederst) for feltet i et plan på tvers av lydaksen, 4,1 cm foran kilden.

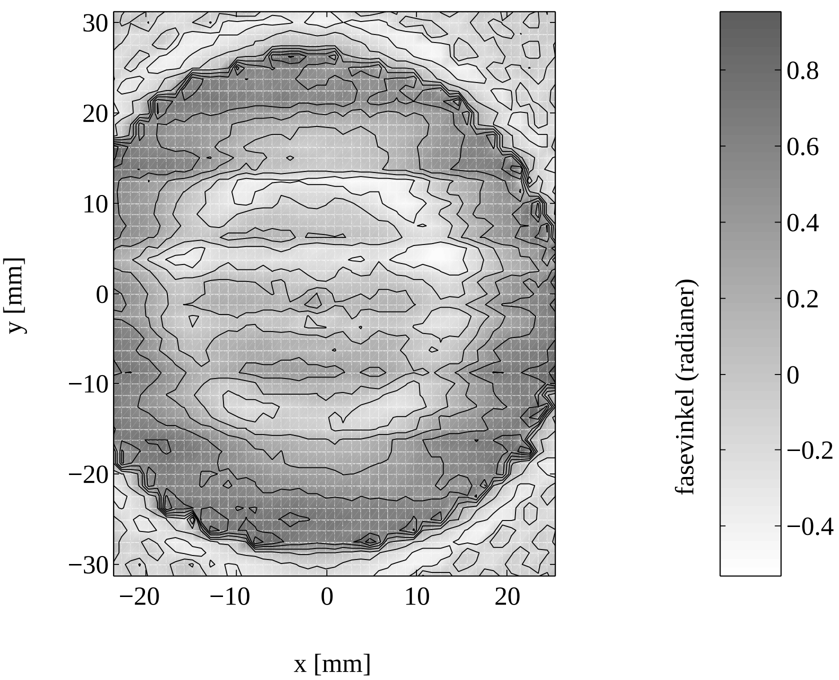


Figur 3.4: Kvartbølgelag, amplituden i lydfeltet i et plan på tvers av lydaksen, 54 cm foran kilden.

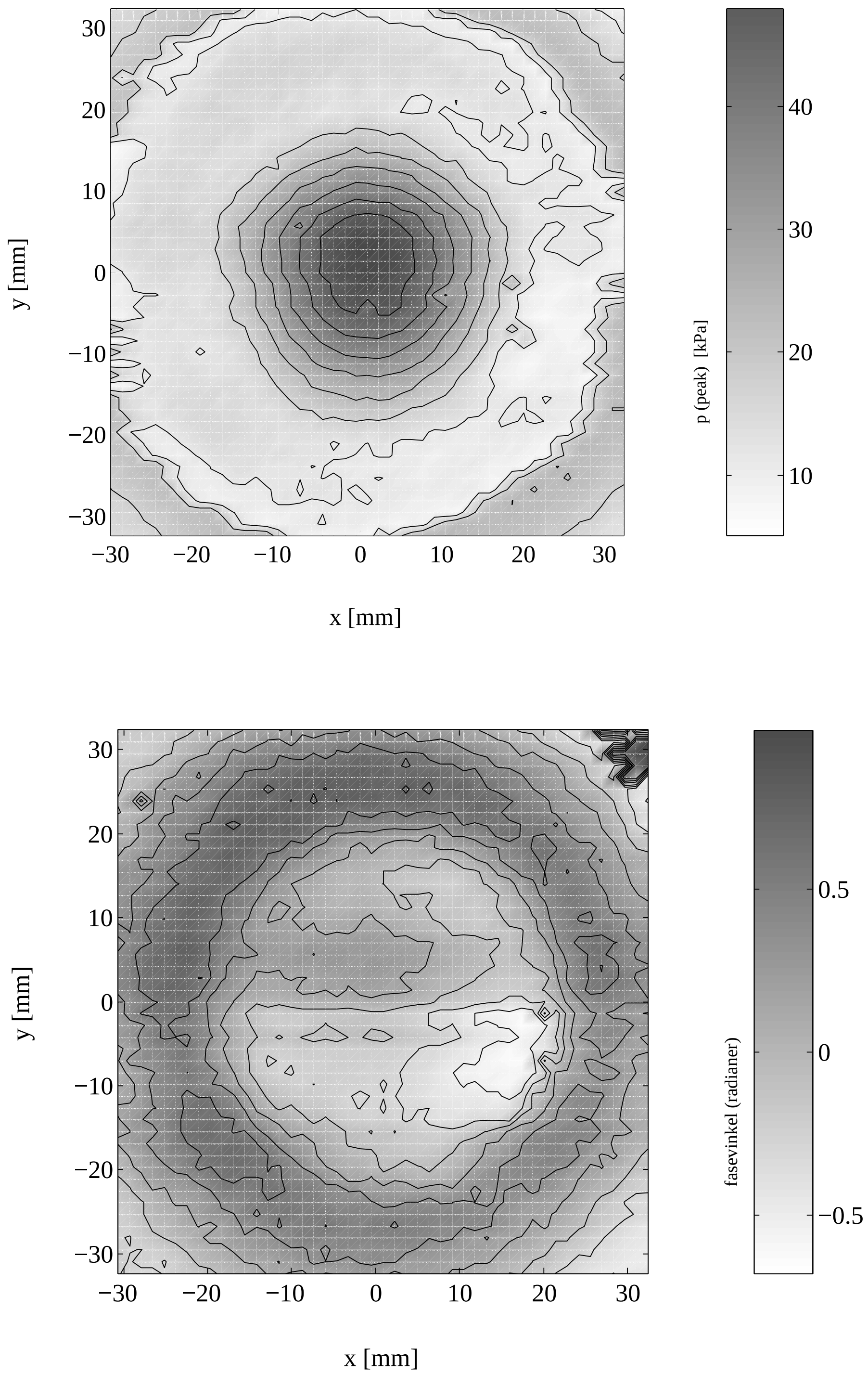


Figur 3.5: Amplitude på tvers av lydaksen. Til venstre: Kvartbølgelag, 4,1 cm ($\cdots$) og 54 cm (—) fra kilden. Til høyre: Trekvartbølgelag, 4,1 cm ($\cdots$) og 75 cm (—) fra kilden.



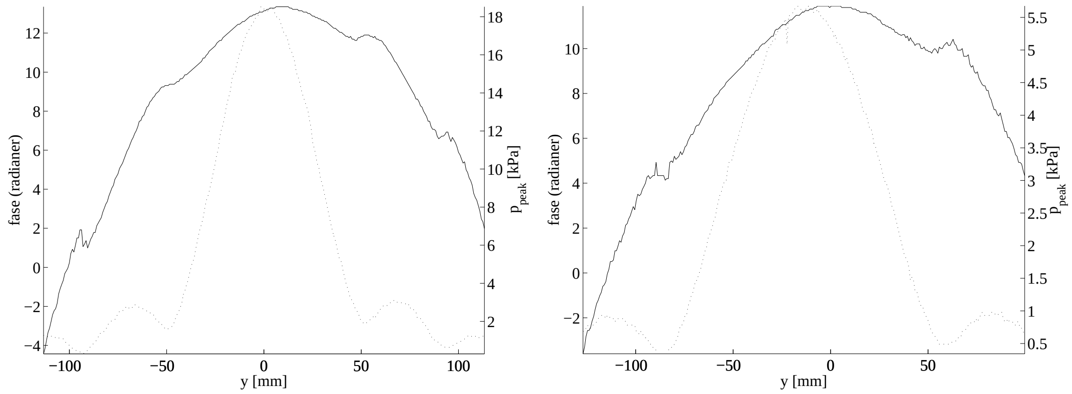


Figur 3.6: 3/4-bølgelag, amplitude (øverst) og fase (nederst) for feltet i et plan på tvers av lydaksen, 4,1 cm foran kilden.



Figur 3.7: 3/4-bølgelag, amplitude (øverst) og fase (nederst) for feltet i et plan på tvers av lydaksen, 75 cm foran kilden.

Bredden på hovedloben i figur 3.8 stemmer godt med direktiviteten som var forventet ut fra posisjonen til siste maksimum langs lydaksen (tabell 3.2). Dette styrker antagelsen om at strålingsfeltet ligner på det fra en plan stempelkilde med radius a_e.



Figur 3.8: Faser på tvers av aksen, målt i tank 2. Til venstre: 149 cm fra lydkilden, til høyre: 214 cm fra kilden. Den stiplede kurven viser trykkamplituden, mens den heltrukne viser fasevariasjonen.

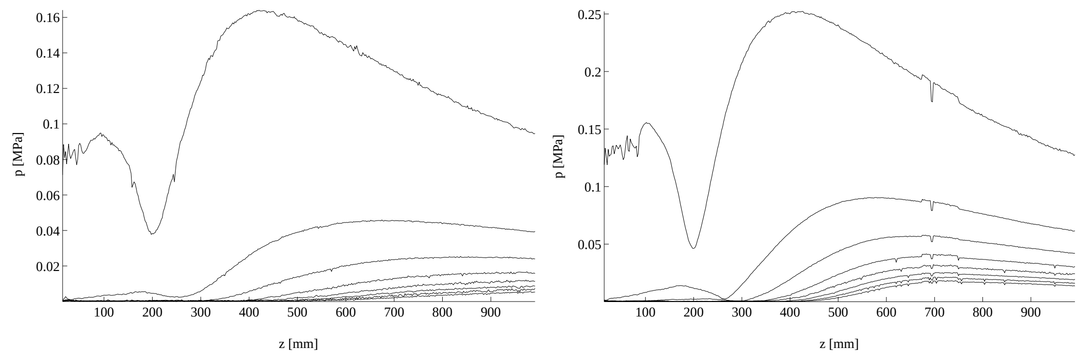
3.2 Ikkelineære lydfelt i vann

3.2.1 Måleresultater

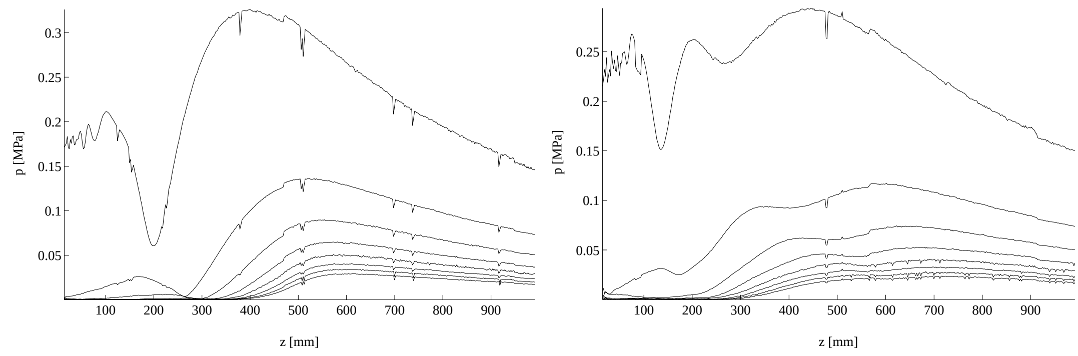
Overføringsfunksjonen H^{Vu} mellom spenning inn på kilden og partikkelhastighet på kildens overflate, ble beregnet ved hjelp av Mason-modellen (`sensf.m`, appendiks A). Den forventede hastighetsamplituden og Mach-tallet $\varepsilon_k = u_k/c_0$ er vist for fire inngangsspenninger i tabell 3.3. Trykket langs aksen ble målt i tank 1 ved disse amplitudene, og de målte bølgeformene ble Fourier-dekomponert. De første åtte harmoniske (grunntonen på 1100 kHz og syv overtoner) er plottet i figur 3.9 og 3.10.

Tabell 3.3: Parametere ved måling av ikkelineære felt: Nominell amplitude inn på effektforsterkeren: V_1. Inngangsspenning på kilden: V_2. Beregnet hastighetsamplitude på kildens overflate (Mason-modellen): u_k. Mach-tall: ε_k.

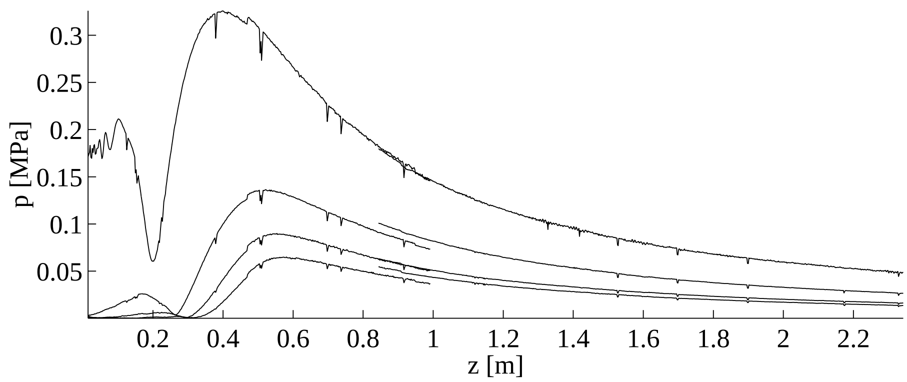
Målesett nr.	V_1 [mV$_{\mathrm{P-P}}$]	V_2 [V$_{\mathrm{P-P}}$]	u_k [m/s]	ε_k [10^{-4}]
1	200	44,4 ± 0,2	0,070±0,003	0,47±0,02
2	300	66,8 ± 0,2	0,105±0,002	0,71±0,02
3	400	90,3 ± 0,3	0,142±0,002	0,95±0,02
4	600	126,2 ± 0,6	0,198±0,003	1,35±0,02



Figur 3.9: Amplituder langs aksen for de 8 første harmoniske, inngangsspenning 200 $mV_{P\text{-}P}$ (venstre) og 300 $mV_{P\text{-}P}$ (høyre).



Figur 3.10: Amplituder langs aksen for de 8 første harmoniske, inngangsspenning 400 $mV_{P\text{-}P}$ (venstre) og 600 $mV_{P\text{-}P}$ (høyre).



Figur 3.11: De fire første harmoniske langs aksen, målt i begge tankene. 400 mV_{P-P} inngangssignal på effektforsterkeren.

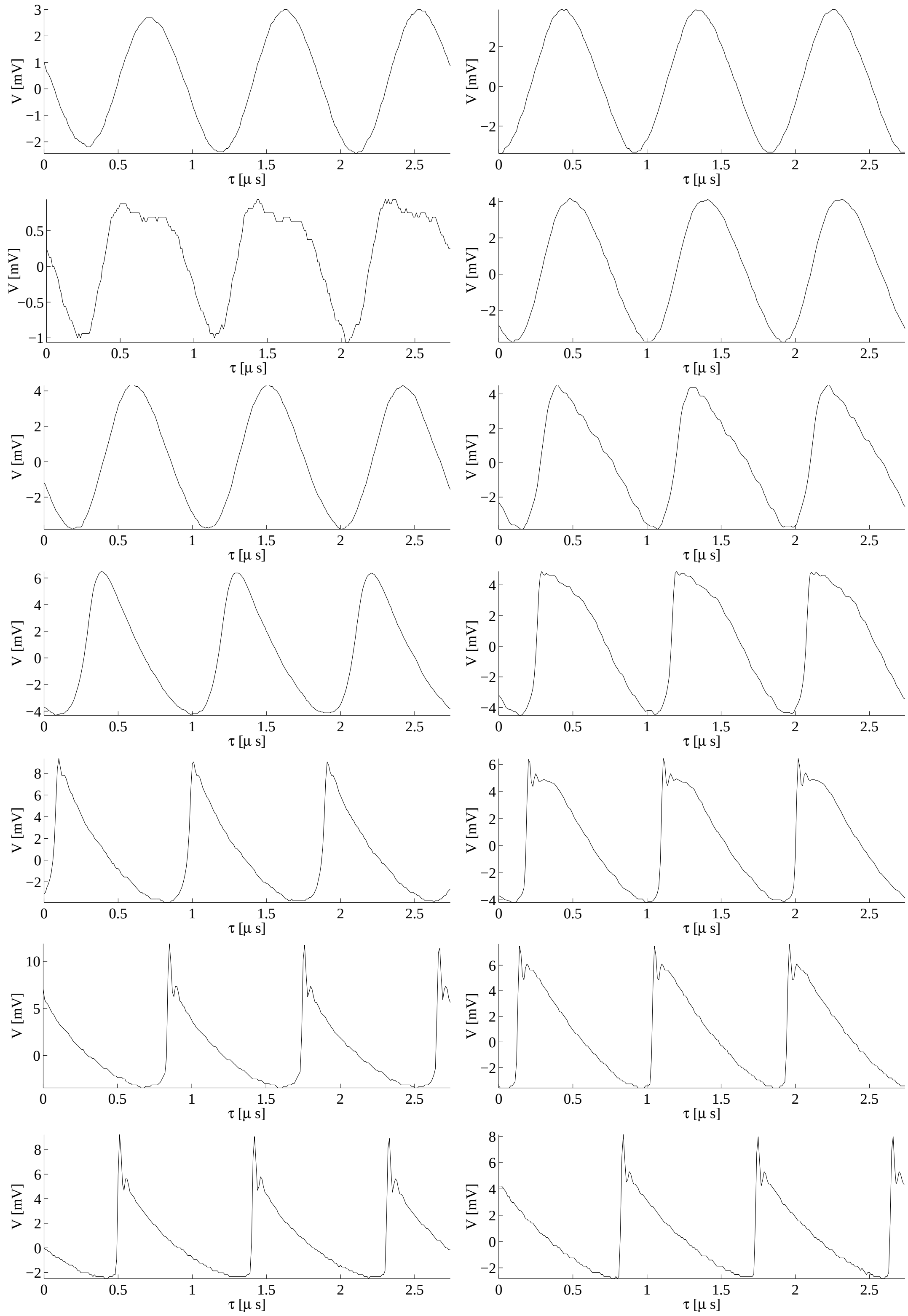
I tank 2 (SOBER-tanken) kunne feltet langs aksen måles i avstander mellom 84 cm og 250 cm fra kilden. Resultatene fra målinger i begge tankene med inngangsamplitude 400 mV_{P-P} på effektforsterkeren, er plottet sammen på figur 3.11.

Trykket langs aksen viser ikke de sterke oscillasjonene som forventes nærmest kilden. Dette kan skyldes at hydrofonens aktive område hadde større utstrekning enn de smale toppene og bunnene i amplituden i denne delen av lydfeltet. Sannsynligheten for å treffe et maksimum eller minimum var i tillegg liten på grunn av avstanden mellom hvert målepunkt.

Måleresultatet for 600 mV_{P-P} viser en annen utvikling langs aksen enn ved de tre lavere amplitudene. Den første overtonen vokser fortere nær kilden, og det oppstår et ekstra minimum i grunntonen som ser ut til å forplante seg til de høyere harmoniske. Ved ca. 70 cm oppnår grunntonen den samme amplituden som når inngangsspenningen på effektforsterkeren er 400 mV_{P-P}.

Figur 3.13 viser at den første sideloben i grunntonen 541 mm fra kilden er høyere ved 600 mV enn ved 400 mV, selv om hovedloben er lavere. Det ser også ut til å være en liten forskjell i sidelobenes avstand fra lydaksen. Bølgeformer på aksen ved 600 mV inngangssignal er sammenlignet med de for 400 mV i figur 3.12. Det er i dette tilfellet ikke kompensert for membranhydrofonens følsomhet. Skarpe sjokk vises derfor som noe større enn i virkeligheten.

Siden måleoppsett 1 ble demontert for å bygge oppsett nummer to, ble det ikke anledning til å undersøke fenomenet nærmere i den første tanken. Resultatet av 19 målinger i tank 2 er vist i figur 3.14. Inngangsspenningen ble variert mellom 300 mV og 600 mV. Grunntonens amplituder på lydaksen er midlet over 200 punkter mellom 84 cm og 126 cm fra kilden. Verdiene er plottet som funksjon av signalamplituden inn på effektforsterkeren. Målingene ble utført med 30 perioder lange pulser og en pulsfrekvens på 200 Hz. Resultatet antyder at sammenhengen mellom den målte amplituden og styrken på inngangssignalet endrer seg



Figur 3.12: Målte bølgeformer langs aksen ved 400 mV_{P-P} (t.v.) og 600 mV_{P-P} (t.h.). Avstandene fra lydkilden er (ovenfra) 52 mm, 200 mm, 300 mm, 400 mm, 500 mm, 600 mm og 800 mm. De plottede verdiene er spenningene som ble målt på membranhydrofonen. Det er ikke korrigert for mottakerfølsomheten. På grunn av en resonans i hydrofonen rundt 20 MHz vises de skarpeste sjokkene som høyere enn de er. Resonansen introduserer også et maksimum nummer to like etter skarpe sjokk.

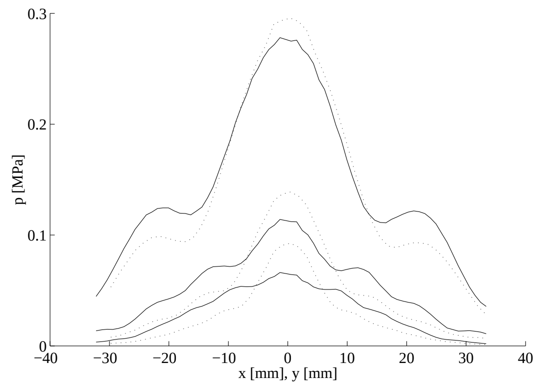
når signalet overstiger ca. 420 mV_{P-P}. De to første overtonene ble undersøkt på samme måte, og ga tilsvarende resultater. For den aktuelle lydstyrken er det bare tilnærmet er riktig å vente en lineær sammenheng mellom trykkamplituden på kilden og grunntonens amplitude et stykke foran den. Det er likevel tegnet inn en rett linje i figuren, for å gi et inntrykk av hvor lydfeltets oppførsel begynner å forandre seg.

Figur 3.14 viser også strøm-amplituden gjennom transduseren som funksjon av spenningen. Også på dette plottet ser man avvik fra en lineær utvikling når spenningen er større enn den som tilsvarer inngangssignaler i området rundt 400 mV_{P-P}. Spenningen og strømmen inn på transduseren ble målt med henholdsvis en LeCroy PP002 1:10-spenningsprobe og en HP 1110B strømprobe. Stømproben hadde en oppgitt nøyaktighet på $\pm 3\%$.

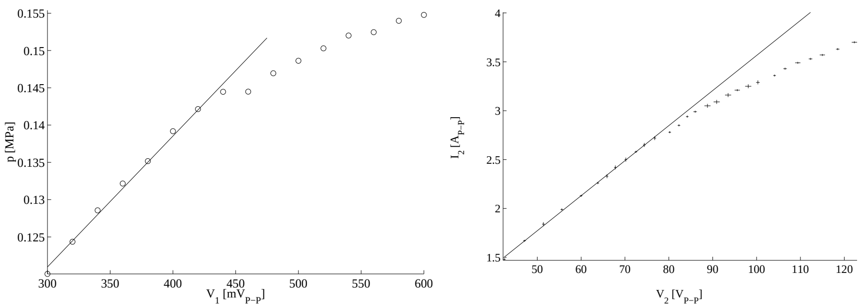
Ved inngangsspenninger høyere enn 400 mV, begynte effektforsterkeren å generere overtoner (3.- og 5.-harmoniske) i utgangssignalet. En piezokeramisk skive kan tilnærmet ses på som en halvbølgeresonator, og de odde harmoniske fra forsterkeren lot seg derfor eksitere i transduseren. At den første av overtonene har vært eksitert kan ses nærmest kilden på det høyre plottet i figur 3.10. Komponenten dør ut i løpet av de første 10 cm. Dette skjer sannsynligvis fordi den er ute av fase med den 3.-harmoniske som genereres av ikkelineariteten i lydstrålen.

Målinger som tilsvarer de som er presentert i figur 3.14, ble gjort med en ENI-A500 60 dB effektforsterker. Denne kan gi større effekt enn forsterkeren som ble brukt ellers i oppgaven. Resultatene er plottet i figur 3.15, og viser at sammenhengen mellom spenning- og strøm-amplitude var tilnærmet lineær i dette tilfellet. Ved de amplitudene som ble forsøkt, ble det ikke observert overtoner i signalet fra den større effektforsterkeren. I strømstyrken gjennom transduseren ble det heller ikke sett andre frekvenskomponenter enn grunntonen. Likevel vokste ikke amplitude n i lydfeltet som forventet når inngangsspenningen på transduseren oversteg ca. 100 V_{P-P}.

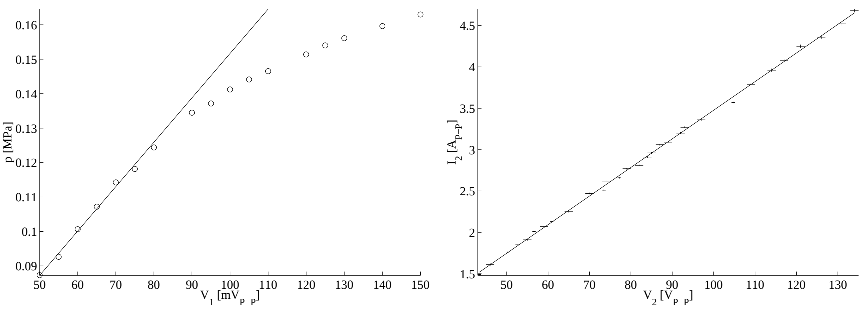
En forklaring på avviket som ble observert, kan være begynnende kavitasjon. Det oppsto kraftig kavitasjon og strømning ved kontinuerlige signaler på ned mot 30 V_{P-P} inn på transduseren. Med de lydpulsene som ble brukt ved målingene, ble det ikke observert tegn på slike effekter. Det er også mulig at måten transduseren svinger på endrer seg ved høyere amplituder, slik at lydfeltet ligner mindre på det fra en stempelkilde.



Figur 3.13: De tre første harmoniske på tvers av lydaksen, 541 mm fra kilden. Stiplet kurve: 400 mV, målt i horisontal retning. Heltrukken kurve: 600 mV, målt i vertikal retning.



Figur 3.14: Måledata fra kildetransduseren og 50 dB-effektforsterkeren (ENI A-240L). Til venstre: Amplitude for grunntonen på aksen, midlet over avstander fra 84 cm til 126 cm (200 punkt). Til høyre: Strøm gjennom transduseren som funksjon av spenningen V_2 over den.



Figur 3.15: Måledata fra kildetransduseren og ENI A-500 effektforsterkeren. Til venstre: Amplitude for grunntonen på aksen, midlet over avstander fra 84 cm til 126 cm (200 punkt). Til høyre: Strøm gjennom transduseren som funksjon av spenningen V_2 over den.

3.2.2 Sammenligning med beregnede verdier

I figur 3.16 er tre av de målte feltene sammenlignet med resultater fra Bergenkoden. De simulerte aksetrykkene ble i hvert tilfelle multiplisert med en faktor K_{sim} (tabell 3.4) for å oppnå samsvar med måleresultatene.

Kildebetingelsen ved løsning av den transformerte bølgeligningen var en plan, sirkulær stempelkilde med radius lik transduserens effektive kilderadius $a_e = 24,3$ mm. Ikkelinearitets-parameteren β ble satt til 3,5, og absorpsjonskoeffisienten som ble brukt var $\alpha_2 = 2,5 \cdot 10^{-14}$ Np/(m Hz^2).

Mach-tallene ε_s som ga de presenterte resultatene fra Bergenkoden, stemmer overens med verdiene ε_k som ble beregnet ved hjelp av Mason-modellen (se tabell 3.3 og 3.4).

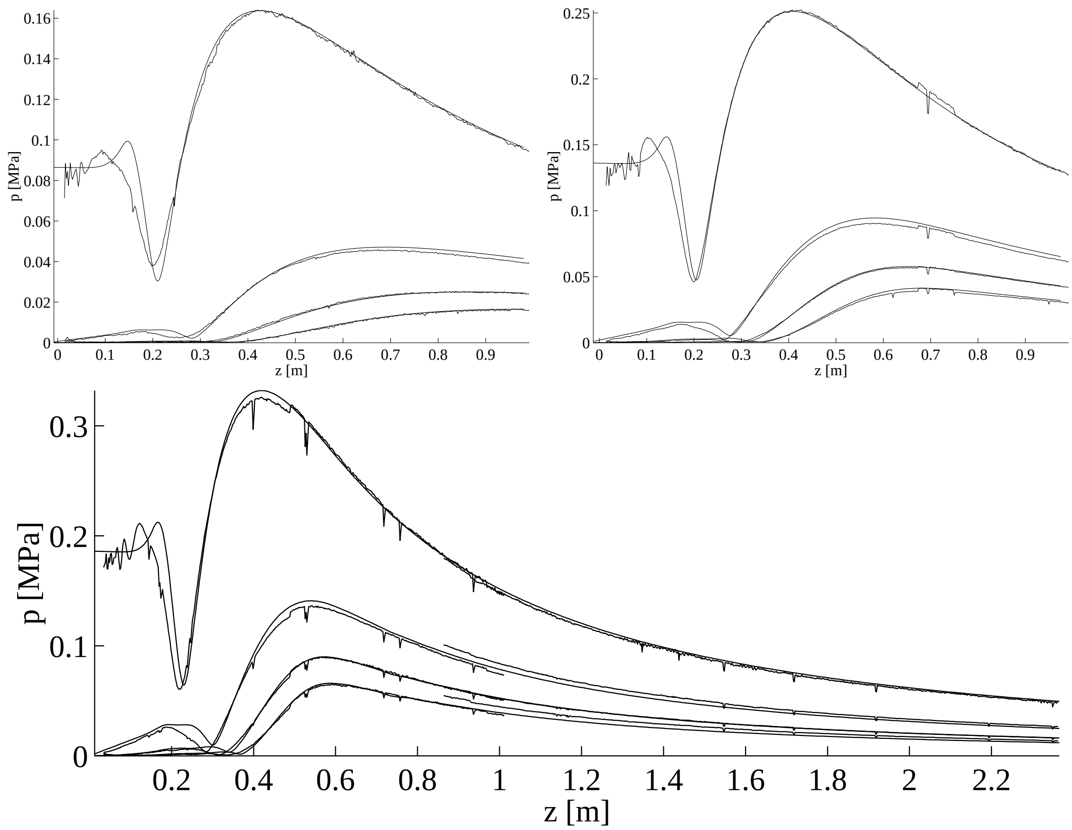
Ved 600 mV$_{\mathrm{P-P}}$ (målesett 4) lyktes det ikke å oppnå samsvar mellom målte og simulerte verdier. Det simulerte feltet utvikler seg på samme måte som for de lavere amplitudene, mens måleresultatet viser en helt annen oppførsel. Når lydfeltet fra transduseren skal sammenlignes med modeller for plane stempelkilder, anbefales det å ikke bruke større spenning enn omkring 90 V$_{\mathrm{P-P}}$.

Tabell 3.4: Parametere brukt ved simulering av aksetrykket: Amplitude inn på effektforsterkeren ved korresponderende målinger: V_1. Mach-tall ved simulering: ε_s. Faktor for tilpasning av resultatet: K_{sim}.

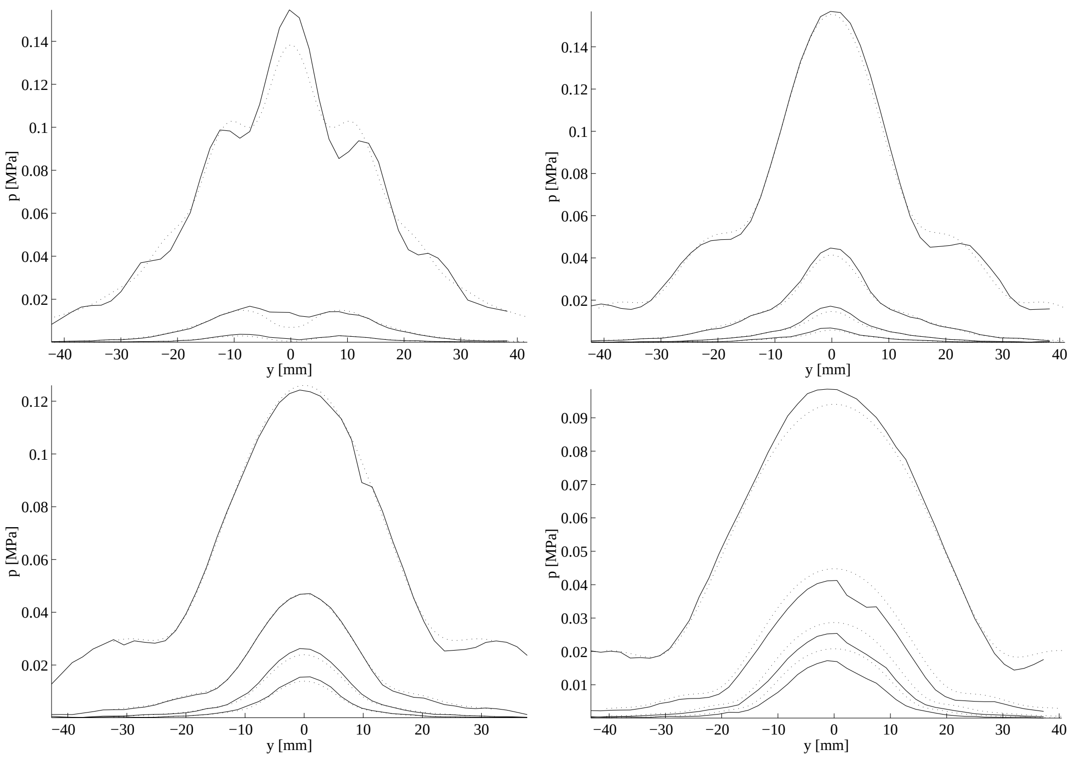
Målesett nr.	V_1 [mV$_{\mathrm{P-P}}$]	ε_s [10^{-4}]	K_{sim}
1	200	0,45	0,873
2	300	0,71	0,871
3	400	0,95	0,889

Som det fremgår av tabell 3.4, var det målte lydtrykket opptil 13% mindre enn det som var forventet ut fra Bergenkoden. Når det beregnede resultatet er multiplisert med faktoren K_{sim}, stemmer utviklingen av høyere harmoniske godt (figur 3.16). Dette tyder på at uoverensstemmelsen kan skyldes måleutstyret. Målefeilen i oscilloskopet ble undersøkt ved hjelp av instrumentets eget kalibreringssignal (firkantsignal med frekvens 1 kHz og peak-to-peak-amplitude 500 mV). Den observerte feilen var i dette tilfellet mindre enn 3%. Oscilloskopet ble også sammenlignet med et av typen Tektronix 2445, uten at det ble funnet noe betydelig avvik. Det ble funnet varierende samsvar mellom resultatene fra de to hydrofonene som ble brukt (se figur 3.11), men det er usikkert om uoverensstemmelsen kan tilskrives målefeilen.

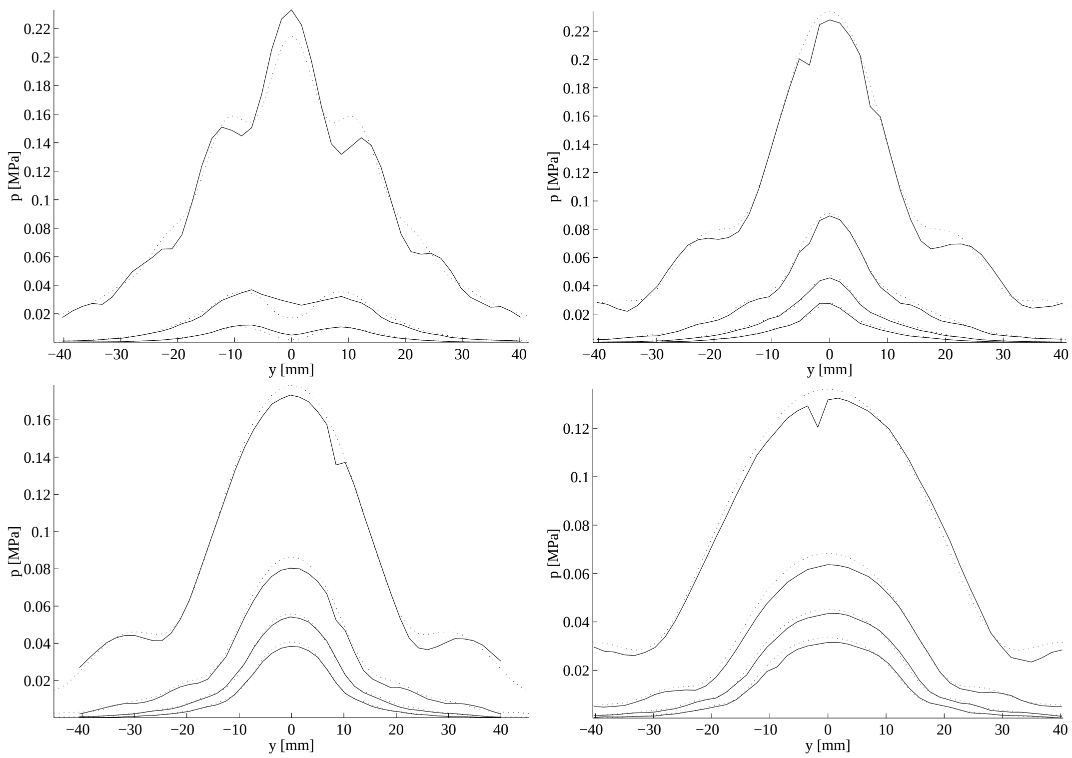
Figur 3.17–3.19 viser transversale målinger sammen med de samme simuleringene som i figur 3.16.



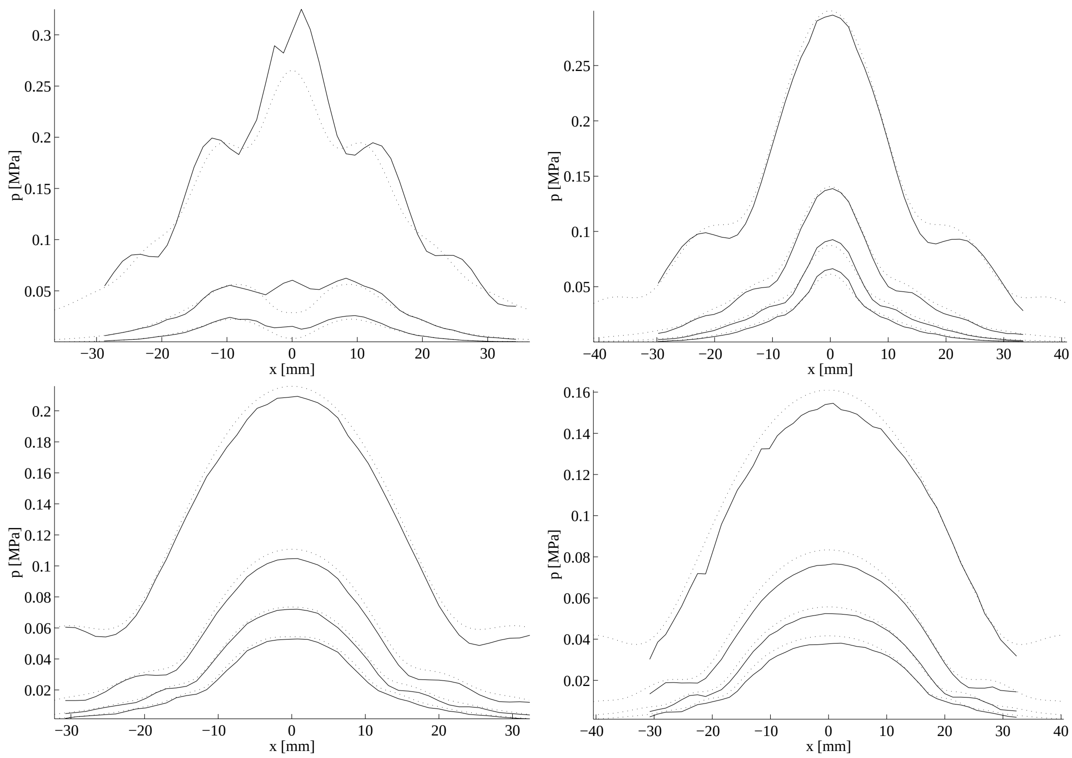
Figur 3.16: De fire første harmoniske langs aksen, målt og beregnet ved hjelp av Bergen-koden. Peak-to-peak inngangsamplituden på effektforsterkeren er henholdsvis 200 mV, 300 mV og 400 mV. Måledataene er de samme som i figur 3.9 og 3.11.



Figur 3.17: De fire første harmoniske på tvers av aksen, målt i tank 1 og beregnet ved hjelp av Bergenkoden. Peak-to-peak inngangsamplitude på effektforsterkeren: 200 mV. Avstandene fra lydkilden var henholdsvis 333 mm, 541 mm, 750 mm og 958 mm.



Figur 3.18: De fire første harmoniske på tvers av aksen, målt i tank 1 og beregnet ved hjelp av Bergenkoden. Peak-to-peak inngangsamplitude på effektforsterkeren: 300 mV. Avstander fra lydkilden, henholdsvis: 333 mm, 541 mm, 750 mm og 958 mm.



Figur 3.19: De fire første harmoniske på tvers av aksen, målt i tank 1 og beregnet ved hjelp av Bergenkoden. Peak-to-peak inngangsamplitude på effektforsterkeren: 400 mV. Avstander fra lydkilden, henholdsvis: 333 mm, 541 mm, 750 mm og 958 mm.

Kapittel 4

Konklusjon

4.1 Oppsummering

En lydkilde ble laget som så ut til å tilfredsstille kravene til driftsfrekvens, direktivitet, utstrålt effekt, transientrespons og justeringsmuligheter. Transduseren ble konstruert med en plan skive av det piezokeramiske materialet Pz 26 som aktivt element. Det ble laget et mekanisk koblingslag mellom skiven og strålingsmediet. For at en høyest mulig kildefølsomhet skulle oppnås, ble det valgt å ha luft på baksiden av elementet.

Egenskapene til den piezokeramiske skiven ble undersøkt, og noen materialparametere ble bestemt ved hjelp av Mason-modellen for tykkelsessvingninger. Modelleringsprogrammet FEMP ble forsøkt brukt til å finne flere av parameterne, men fremgangsmåten som ble valgt viste seg å være usikker.

Mason-modellen ble videre brukt til å spesifisere egenskapene som var ønsket for det mekaniske koblingslaget. Dette ble støpt direkte på transduseren av en blanding av epoxy-støpemasse og kobberpulver. Blandingsforholdet bestemte materialets spesifikke akustiske impedans, og ble beregnet ved hjelp av Woods modell. To koblingslag med ulik tykkelse og impedans ble prøvd på lydkilden.

Den ferdige transduseren ble testet i to måleoppsett. Kildens transientresponser ble undersøkt, og det ble gjort målinger i strålingsfeltet ved ulike amplituder.

Bergenkoden ble brukt til løsning av KZK-ligningen for en plan, sirkulær stempelkilde med radius lik transduserens effektive kilderadius. Det ble funnet et systematisk avvik på opptil 13% mellom målte og beregnede amplituder. Ved å multiplisere resultatet fra Bergenkoden med en korreksjonsfaktor, ble det oppnådd samsvar med måleresultatene opp til en bestemt amplitude. Over denne amplituden endret det målte feltet oppførsel, og modellen beskrev det ikke lenger på en tilfredsstillende måte.

4.2 Forslag til videre arbeid

I denne oppgaven ble det forsøkt å bestemme materialparametere ved hjelp av FEMP. Dette ble gjort for å undersøke hvor presise de oppgitte verdiene var, og fordi en endelig-element-simulering kunne være til hjelp under konstruksjonen av lydkilden. Det viste seg at FEMP konvergerte forholdsvis langsomt for visse parametersett og geometrier, og dermed krevde for mye dataressurser til at beregningene kunne utføres på en vanlig PC. Relevante temaer i forbindelse med endelig-element-metoden kan være utvikling av sikrere fremgangsmåter for bestemmelse av materialkonstanter, i tillegg til nærmere kartlegging av konvergensegenskapene til FEMP. Det kan også være interessant å oversette FEMP til et språk som håndterer minneressurser på en annen måte enn Matlab.

FEMP kan beregne partikkelforskyvningen på overflaten av en simulert transduserkonstruksjon. Ved å bruke slike resultater i kildebetingelser til Bergenkoden, vil man kanskje kunne oppnå mer helhetlige simuleringer som vil være til nytte ved konstruksjon av lydkilder.

Det utstrålte feltet fra transduseren som ble konstruert i denne oppgaven, endrer seg på en uventet måte når inngangsspenningen overstiger ca. 100 V_{P-P}. Om endringen skyldes effekter i strålingsmediet eller i lydkilden, og på hvilken måte, er ennå ikke klarlagt.

Appendiks A — Programlisting

Bergenkoden

tbe_plan_fok.f

Kode for løsning av den transformerte bølgeligningen (TBE). Dette programmet ble brukt i oppgaven.

```
      PROGRAM BEAM
      INTEGER IXMAX,NHARM,NUMBE,t,antall
      PARAMETER (IXMAX=4000,NHARM=60,NUMBE=5)
      DOUBLE PRECISION OMEGA(0:IXMAX,NHARM,2),WINDWX,START,STOP,
     +                 WBEAM(NUMBE),COLD(NHARM),CNEW(NHARM),
     +                 KHALF(NHARM),NONLI(0:IXMAX,NHARM,2),
     +                 A(2,2, - 1:1,0:IXMAX),W(2,0:IXMAX),
     +                 F(2,0:IXMAX),WORK(12*IXMAX+12),
     +                 FREQ,RADIUS,SSPEED,ALFA2,BETA,MACH,PI,
     +                 r0,rd,k,d
      EXTERNAL COEFF,NONLIN,STEPP,BEWRI
      DOUBLE PRECISION PABSOR,PNONLI,ISSTEP,fl
      CHARACTER*10 transv,ax,desc,fn
      COMMON /PARAM/PABSOR,PNONLI
      COMMON /AXSTEP/ISSTEP
      common /focusing/k,fl,radius

      fn = 'inn1' ! Name for the parameter file
      antall = 3 ! Number of lines in the parameter file
      OPEN(UNIT=9,FILE=fn,STATUS="unknown")
C     Først tre filnavn, så parametere
        write(9,2) 'trans1','akse1','desc1',
     +             'M = ',0.70d-4,'a = ',24.35d-3,
     +             'beta = ',3.5
        write(9,2) 'trans2','akse2','desc2',
     +             'M = ',0.72d-4,'a = ',24.35d-3,
     +             'beta = ',3.5
        write(9,2) 'trans3','akse3','desc3',
     +             'M = ',0.74d-4,'a = ',24.35d-3,
     +             'beta = ',3.5
      CLOSE(9)
C
2     FORMAT(a6,1x,a5,1x,a5,2x,a4,e16.10,2x,a4,e16.10,2x,
     +       a7,e16.10)
3     FORMAT(a6,1x,a5,1x,a5,6x,e16.10,6x,e16.10,9x,e16.10)
C     Innfil fra matlab (inversjon.m)
c3     format(a4,7x,e16.10,7x,e16.10,10x,e16.10)
C
C     Run the program for each line of fn
C
      OPEN(UNIT=9,FILE=fn,STATUS='old')
      DO 5 t=1,antall
        READ(9,3) transv,ax,desc,mach,radius,beta
        OPEN (UNIT=10,FILE=transv,STATUS='unknown')
        OPEN (UNIT=11,FILE=ax,STATUS='unknown')
        OPEN (unit=12,file=desc,STATUS='unknown')
C
C       Read input (For beregning av PABSOR og PNONLI)
C
c
C       Frekvens  Hz
        FREQ = 1.1d6
C       Radius, m (X = x/RADIUS)
c        RADIUS = 23.D-3
C       Lydhastighet, m/s
        sspeed = 1490.0
C       Absorbsjonskonstant, s^2/m
        alfa2 = 2.5d-14
C       Ikkelinearitetsparameter
c        beta = 3.5
C       Machtall
c        mach = 1.2D-4
C       Bredde paa sinus-avrunding av kilden
        d = 0.
C       Fokallengde
C        fl = 2.
C
        PI = 4*ATAN(1.0D0)
        k  = 2*pi*freq/sspeed
        r0 = radius**2*k/2
        rd = 1/(beta*mach*k)
        PABSOR = alfa2*freq**2*r0
        PNONLI = r0/rd
c
c
        WINDWX = 70.
        ISSTEP = 3.5D-3
        wbeam(1) = 0.333/r0
        wbeam(2) = 0.541/r0
        wbeam(3) = 0.75/r0
        wbeam(4) = 0.958/r0
        wbeam(5) = 1.d0/r0
c
C       Initiate the Fourier coefficients.
c
        CALL INIT(WINDWX,OMEGA,IXMAX,NHARM,d)
C
C       Integrate the differential equations.
C
        START = 0
        STOP = WBEAM(NUMBE)
C
        WRITE(12,*) 'c = ',sspeed,' a = ',radius,' f = ',freq
        WRITE(12,*) 'pabsorb = ',pabsor,' pnonlin = ',pnonli
        WRITE(12,*) 'Machtall = ',mach,' Rayl-dist = ',r0,
     + ' Sjokkavst = ',rd
        WRITE(12,*) 'Snittposisjoner: ',(wbeam(i),i=1,numbe)
        WRITE(12,*) 'Antall harmoniske: ',nharm,'d = ',d
        CLOSE(12)
C
        WRITE(6,*) 'kzkcyl, M =',mach,', d = ',d,',
     +             nharm = ',nharm
        CALL KZKCYL(OMEGA,IXMAX,NHARM,WINDWX,START,STOP,WBEAM,
     +              NUMBE,COEFF,NONLIN,STEPP,BEWRI,COLD,CNEW,
     +              KHALF,NONLI,A,W,F,WORK)
      CLOSE(10)
      CLOSE(11)
      CLOSE(12)
5     CONTINUE
      CLOSE(9)
      STOP
      END

      SUBROUTINE INIT(WINDWX,OMEGA,IXMAX,NHARM,d)
      INTEGER I,J,IXMAX,NHARM
      DOUBLE PRECISION OMEGA(0:IXMAX,NHARM,2),WINDWX,d
      DOUBLE PRECISION USTEP,FA,X
      DO 20 I = 0,IXMAX
          IF (NHARM.GT.1) THEN
              DO 10 J = 2,NHARM
                  OMEGA(I,J,1) = 0
                  OMEGA(I,J,2) = 0
10            CONTINUE
          END IF
          X = I* WINDWX/DBLE(IXMAX+1)
          OMEGA(I,1,1) = USTEP(X,d)*SIN(FA(X))
          OMEGA(I,1,2) = USTEP(X,d)*COS(FA(X))
20    CONTINUE
      RETURN
      END
C
C     p(U,t) = A cos(wt) + B sin(wt)
C            = USTEP(X)*sin(wt + FA(X))
C
      DOUBLE PRECISION FUNCTION FA(U)
      DOUBLE PRECISION U,k,fl
      common /focusing/k,fl,radius
C      fa = -k*radius**2*(abs(u))**2/(2*fl)
      FA = (ABS(U))**2
      RETURN
      END
      DOUBLE PRECISION FUNCTION USTEP(U,d)
      DOUBLE PRECISION U,d,pi
      IF (abs(U).le.(1.d0 - d/2)) THEN
          USTEP = 1
      ELSE
          if (abs(U).le.(1.d0 + d/2)) then
              pi = 4*atan(1.d0)
              USTEP = 0.5*(1 + sin(pi/d*(1 - abs(U))))
          else
              USTEP = 0
          end if
      END IF
      RETURN
      END
      SUBROUTINE BEWRI(WINDWX,SIGMA,OMEGA,IXMAX,NHARM)
      INTEGER NHARM,IXMAX,J,I
      DOUBLE PRECISION SIGMA,OMEGA(0:IXMAX,NHARM,2),WINDWX
      DOUBLE PRECISION S,X,AMP,AMPL,DB,AMPAX,amp2,db2,ampax2
      DOUBLE PRECISION amp3,amp4,amp5
      S = (1.D0+SIGMA)
      DO 20 J = 1,1
          DO 10 I = 0,IXMAX
```

```
            X = I*WINDWX/DBLE(IXMAX+1)
            AMP = AMPL(OMEGA(I,J,1),OMEGA(I,J,2),S)
            AMP2 = AMPL(OMEGA(I,2,1),OMEGA(I,2,2),S)
            AMP3 = AMPL(OMEGA(I,3,1),OMEGA(I,3,2),S)
            AMP4 = AMPL(OMEGA(I,4,1),OMEGA(I,4,2),S)
            AMP5 = AMPL(OMEGA(I,5,1),OMEGA(I,5,2),S)
           WRITE (10,22) SNGL(X),SNGL(amp),SNGL(amp2),
     +                   SNGL(amp3),SNGL(amp4),SNGL(amp5)
10        CONTINUE
20    CONTINUE
22    FORMAT(6(e16.10,1x))
      WRITE(6,*) 'transv updated'
      RETURN
      END
C
C     On the axis:
C
      SUBROUTINE BEWRI2(SIGMA,OMEGA,IXMAX,NHARM)
      DOUBLE PRECISION SIGMA,OMEGA(0:IXMAX,NHARM,2)
      DOUBLE PRECISION AMP(5),AMPL,S
      INTEGER NHARM,IXMAX,I
      S = (1.D0+SIGMA)
      DO 25 I = 1,5
           AMP(I) = AMPL(OMEGA(0,I,1),OMEGA(0,I,2),S)
25    CONTINUE
      WRITE(11,28) sngl(SIGMA),(AMP(I),I=1,5)
28    FORMAT(6(e16.10,1x))
      RETURN
      END
      SUBROUTINE NONLIN(SIGMA,OMEGA,IXMAX,NHARM,NONLI)
      INTEGER NHARM,IXMAX,I,J,N
      DOUBLE PRECISION OMEGA(0:IXMAX,NHARM,2)
      DOUBLE PRECISION NONLI(0:IXMAX,NHARM,2)
      DOUBLE PRECISION SIGMA,FAKT,S,PABSOR,PNONLI
      COMMON /PARAM/PABSOR,PNONLI
      S = (1.D0+SIGMA)
      DO 100 N = 1,NHARM
          DO 29 I = 0,IXMAX
              NONLI(I,N,1) = 0
              NONLI(I,N,2) = 0
29        CONTINUE
          DO 50 I = 0,IXMAX
              DO 30 J = 1,N - 1
                  NONLI(I,N,1) = NONLI(I,N,1) +
     +                           (OMEGA(I,J,2)*OMEGA(I,N-J,1))
                  NONLI(I,N,2) = NONLI(I,N,2) +
     +                           OMEGA(I,J,2)*OMEGA(I,N-J,2) -
     +                           OMEGA(I,J,1)*OMEGA(I,N-J,1)
30            CONTINUE
              NONLI(I,N,2) = 0.5*NONLI(I,N,2)
              DO 40 J = N + 1,NHARM
                  NONLI(I,N,1) = NONLI(I,N,1) +
     +                           OMEGA(I,J-N,1)*OMEGA(I,J,2) -
     +                           OMEGA(I,J-N,2)*OMEGA(I,J,1)
                  NONLI(I,N,2) = NONLI(I,N,2) -
     +                           OMEGA(I,J-N,1)*OMEGA(I,J,1) -
     +                           OMEGA(I,J-N,2)*OMEGA(I,J,2)
40            CONTINUE
50        CONTINUE
          FAKT = PNONLI*N/ (2*S)
          DO 60 I = 0,IXMAX
              NONLI(I,N,1) = FAKT*NONLI(I,N,1)
              NONLI(I,N,2) = FAKT*NONLI(I,N,2)
60        CONTINUE
100   CONTINUE
      RETURN
      END
      DOUBLE PRECISION FUNCTION AMPL(FA,FB,S)
C
C     FUNCTION CALCULATING AMPLITUDE.
C
      DOUBLE PRECISION FA,FB,S,SD
      SD = FA**2 + FB**2
      IF (S.NE.0) THEN
          AMPL = DSQRT(SD)/S
      ELSE
          AMPL = DSQRT(SD)
      END IF
      END
      SUBROUTINE COEFF(SIGC,SIGK,C,K,NHARM)
      INTEGER NHARM,J
      DOUBLE PRECISION SIGC,SIGK,C(NHARM),K(NHARM)
      DOUBLE PRECISION PABSOR,PNONLI,SC,SK
      COMMON /PARAM/PABSOR,PNONLI
      SC = (1+SIGC)
      SK = (1+SIGK)
      DO 10 J = 1,NHARM
          C(J) = PABSOR* (J**2)
          K(J) = - 1.0/ (4*J* (SK**2))
10    CONTINUE
      RETURN
      END
      DOUBLE PRECISION FUNCTION STEPP(SIGMA)
      DOUBLE PRECISION SIGMA,ISSTEP
      COMMON /AXSTEP/ISSTEP
      STEPP = ISSTEP*(1 + SIGMA)**2
      RETURN
      END
      SUBROUTINE KZKCYL(OMEGA,IXMAX,NHARM,WINDWX,START,STOP,
     +                  WBEAM,NUMBE,COEFF,NONLIN,STEPP,BEWRI,
     +                  COLD,CNEW,KHALF,NONLI,A,W,F,WORK)
C***BEGIN PROLOGUE KZKCYL
C***AUTHOR Jarle Berntsen, University of Bergen,
C         EDB-senteret, Herman Fossgt 6, 5007 Bergen, NORWAY.
C***PURPOSE KZKCYL computes approximations to the Khokhlov-
C          Zabolotzkaya-Kuznetsov equation,see [1,2].
C          After a Fourier-series expansion of the pressure,
C          we may compute approximations to the KZK equation by
C          solving a system of nonlinear parabolic equations of
C          the form:
C
C       dg(n)/dsigma = -c(n,sigma)g(n)
C                      + k(n,sigma)Laplacian(h(n))
C                      + il1(n,sigma,g,h)
C
C       dh(n)/dsigma = -c(n,sigma)h(n)
C                      - k(n,sigma)Laplacian(g(n))
C                      + il2(n,sigma,g,h)
C
C          The terms on the right hand side of the equality sign
C          are due to (from left to right) absorbtion,
           diffraction and nonlinearity.
C          In KZKCYL the Laplacian is cylindrical coordinates.
C
C***DESCRIPTION Two numerical techniques are applied in KZKCYL in
C              order to compute approximations to the system of
C              equations given above.
C              For uniform sources which are of particular
C              interest, Gibbs oscillations appear in the
C              numerical solution. In order to damp these we
C              first take a few steps with a fully implicit
C              method.
C              In order to control the damping
C              the S first harmonics of the Fourier expansion
C              of the initial values are kept to 99.9 per cent
C              of their full strength.
C              S is set as a parameter in KZKCYL.
C
C              After having removed the Gibbs oscillations we
C              continue with the Trapezoidal rule (or
C              Crank-Nicolson method or on this particular
C              problem, Richtmyers procedure) which has 2'nd
C              order accuracy and is unconditionally stable.
C
C              The linear systems of equations that have to be
C              solved in each step are solved with the routine
C              TRIDIA.
C
C   ON ENTRY
C
C     OMEGA  Real array of dimension (0:IXMAX,NHARM,2).
C            Contains in each step approximative values to g(n)
C            and h(n).
C            The first index defines the values of x for which
C            g(n) and h(n) are approximated.
C            The second index defines the harmonic number.
C            If the last index is 1, approximations to g(n) are
C            given. If the last index is 2, approximations to h(n)
C            are given. On entry OMEGA must specify the initial
C            values of g(n) and h(n).
C
C     IXMAX  Integer.
C            Defines the number of interior grid points.
C
C     NHARM  Integer.
C            Defines the number of harmonics retained in the
C            numerical solution.
C
C     WINDWX Real.
C            WINDWX is the upper limit of x.
C
C     START  Real.
C            The starting value of sigma.
C
C     STOP   Real.
C            The final value of sigma.
```

```
C
C     WBEAM  Real array of dimension NUMBE.
C            Specifies the values of sigma where we want
C            print out of OMEGA.
C
C     NUMBE  Integer.
C            The length of the array WBEAM.
C
C     COEFF  Subroutine, supplied by the user.
C            COEFF must be declared as external in the calling
C            program. COEFF is called by KZKCYL to compute the
C            coefficients c(n,sigma) and k(n,sigma).
C
C            Its specification is:
C            SUBROUTINE COEFF(SIGC,SIGK,C,K,NHARM)
C            INTEGER NHARM
C            DOUBLE PRECISION SIGC,SIGK,C(NHARM),K(NHARM)
C
C            SIGC defines on entry the value of sigma for which
C            you want to compute the coefficients c(n,sigma).
C
C            SIGK defines on entry the value of sigma for which
C            you want to compute the coefficients k(n,sigma).
C
C            C contains on exit the
C            values of c(n,SIGC), n=1,...,NHARM.
C
C            K contains on exit the
C            values of k(n,SIGK), n=1,...,NHARM.
C
C     NONLIN Subroutine, supplied by the user.
C            NONLIN must be declared as external in the calling
C            program. NONLIN is called by KZKCYL to compute the
C            nonlinear terms il1(n,sigma,g,h) and
C            il2(n,sigma,g,h).
C
C            Its specification is:
C            SUBROUTINE NONLIN(SIGMA,OMEGA,IXMAX,NHARM,NONLI)
C            INTEGER IXMAX,NHARM
C            DOUBLE PRECISION SIGMA
C            DOUBLE PRECISION OMEGA(0:IXMAX,NHARM,2)
C            DOUBLE PRECISION NONLI(0:IXMAX,NHARM,2)
C
C            SIGMA defines on entry the value of sigma for which
C            you want to compute the nonlinear terms.
C
C            OMEGA defines on entry the computed approximations
C            to g(n) and h(n).
C
C            NONLI must on exit contain the nonlinear terms.
C            If the last index is 1, approx. to il1 is indicated.
C            If the last index is 2, approx. to il2 is indicated.
C
C     STEPP  Real function, supplied by the user.
C            STEPP must be declared as external in the calling
C            program. STEPP defines the step size in sigma
C            direction.
C
C
C            Its specification is:
C            DOUBLE PRECISION FUNCTION STEPP(SIGMA)
C            DOUBLE PRECISION SIGMA
C
C     BEWRI  Subroutine, supplied by the user.
C            BEWRI must be declared as external in the calling
C            program. BEWRI is called by KZKCYL to write
C            information in OMEGA to file(s).
C
C            Its specification is:
C            SUBROUTINE BEWRI(WINDWX,SIGMA,OMEGA,IXMAX,NHARM)
C            INTEGER IXMAX,NHARM
C            DOUBLE PRECISION WINDWX,SIGMA
C            DOUBLE PRECISION OMEGA(0:IXMAX,NHARM,2)
C
C  ON RETURN
C
C     OMEGA  contains the approximations to g(n) and h(n) at
C            sigma = STOP.
C
C     COLD   Real work array of dimension NHARM.
C
C     CNEW   Real work array of dimension NHARM.
C
C     KHALF  Real work array of dimension NHARM.
C
C     NONLI  Real work array of dimension (0:IXMAX,NHARM,2).
C
C     A      Real work array of dimension (2,2,-1:1,0:IXMAX).
C
C     W      Real work array of dimension (2,0:IXMAX).
C
C     F      Real work array of dimension (2,0:IXMAX).
C
C     WORK   Real work array of dimension (12*IXMAX+12).
C
C***REFERENCES
C     [1]Zabolotskaya,E.A. and Khokhlov,V.Pa,
C        Quasi-plane waves in the nonlinear acoustics of confined
C        beams, Sov.Phys.Acoust.,15,1969,pp.35-40.
C     [2]Kuznetsov,V.P.,Equations of nonlinear acoustics,
C        Sov.Phys.Acoust.,16,1971,pp.467-470.
C     [3]Hamilton,M.F. and Naze Tjotta,J. and Tjotta.S,
C        Nonlinear effects in the farfield of a directive
C        sound source,
C        J.Acoust.Soc.Am.,78,1985,pp.202-216.
C     [4]Berntsen,J.,The numerical calculation of the Khokhlov-
C        Zabolotskaya-Kuznetsov equation for piston sources,
C        Unpublished paper,1988.
C***ROUTINES CALLED COEFF,NONLIN,STEPP,BEWRI,TRIDIA
C***END PROLOGUE KZKCYL
C
C   Global variables.
C
      INTEGER IXMAX,NHARM,NUMBE
      DOUBLE PRECISION OMEGA(0:IXMAX,NHARM,2),WINDWX,START,STOP,
     +                 WBEAM(NUMBE),STEPP,COLD(NHARM),CNEW(NHARM),
     +                 KHALF(NHARM),NONLI(0:IXMAX,NHARM,2),
     +                 A(2,2, - 1:1,0:IXMAX),W(2,0:IXMAX),
     +                 F(2,0:IXMAX),WORK(12*IXMAX+12)
      EXTERNAL COEFF,NONLIN,STEPP,BEWRI
C
C   Local variables.
C
      INTEGER L,I,BCOUNT,S,IMPSTP,IS,I1,I2
      DOUBLE PRECISION STEP,SIG,OLDSIG,SIGMA,R,DIAG,OFFDIA,
     +                 PI,XSTEP
      DOUBLE PRECISION SIGC,SIGK,FAKT1,FAKT2
      REAL wax,dwax
C
C   IMPSTP is the number of initial fully implicit steps.
C   The strength of the S first harmonics in the Fourier
C   expansion of the initial values must not be damped more
C   than 0.1 percent when taking the IMPSTP initial steps.
C
      PARAMETER (IMPSTP=5)
      PARAMETER (S=20)
C
C***FIRST EXECUTABLE STATEMENT KZKCYL
C
      PI = 4*ATAN(1.d0)
      BCOUNT = 1
      SIGMA = START
      dwax = 0.001
      wax = START + dwax
C
C   By setting START = STOP, we may print out the initial values.
C
      IF (START.EQ.STOP) THEN
         CALL BEWRI(WINDWX,SIGMA,OMEGA,IXMAX,NHARM)
         GO TO 999
      END IF
      XSTEP = ABS(WINDWX)/DBLE(IXMAX+1)
      SIGC = SIGMA
      SIGK = SIGMA
      CALL COEFF(SIGC,SIGK,COLD,KHALF,NHARM)
C
C   Go IMPSTP steps with the fully implicit method.
C
      DO 100 IS = 1,IMPSTP
C
C   Choose the step size such that the S first eigenvectors are
C   kept to 99.9 per cent of their initial strength after IMPSTP
C   steps.
         STEP = SQRT(2./ (1000.*IMPSTP))* (2*WINDWX)**2/
     +          (ABS(KHALF(1))*S**2*PI**2)

         step = step/45    ! 06.03.02

         SIG = SIGMA + STEP
         OLDSIG = SIGMA
         IF (BCOUNT.LE.NUMBE) THEN
            SIGMA = MIN(SIG,WBEAM(BCOUNT),STOP)
         ELSE
            SIGMA = STOP
         END IF
         STEP = SIGMA - OLDSIG
```

```
C
C   Compute the coefficients and the nonlinear
C   contributions.
C
          SIGC = SIGMA
          SIGK = SIGMA
          CALL COEFF(SIGC,SIGK,CNEW,KHALF,NHARM)
          CALL NONLIN(SIGMA,OMEGA,IXMAX,NHARM,NONLI)
          R = STEP/XSTEP**2
C
C   Loop for each harmonic L.
C
          DO 50 L = 1,NHARM
C
C   Set up the appropriate systems of equations in the
C   way required for TRIDIA.
C
             DIAG = 1 + STEP*CNEW(L)
             OFFDIA = 2*R*KHALF(L)
C
C   First the diagonal 2*2 sub-matrices.
C
             A(1,1,0,0) = DIAG
             A(1,2,0,0) = 2*OFFDIA
             A(2,1,0,0) = - A(1,2,0,0)
             A(2,2,0,0) = DIAG
             DO 10 I = 1,IXMAX
                A(1,1,0,I) = DIAG
                A(1,2,0,I) = OFFDIA
                A(2,1,0,I) = - OFFDIA
                A(2,2,0,I) = DIAG
10           CONTINUE
C
C   Then the sub-matrices above the diagonal.
C
             A(1,2,1,0) = - 2*OFFDIA
             A(2,1,1,0) = - A(1,2,1,0)
             DO 20 I = 1,IXMAX - 1
                A(1,2,1,I) = - OFFDIA* (1+1/DBLE(2*I))/2
                A(2,1,1,I) = - A(1,2,1,I)
20           CONTINUE
C
C   and the sub-matrices below the diagonal.
C
             DO 30 I = 1,IXMAX
                A(1,2, - 1,I) = - OFFDIA* (1-1/DBLE(2*I))/2
                A(2,1, - 1,I) = - A(1,2, - 1,I)
30           CONTINUE
C
C   Define the right hand side.
C
             DO 40 I = 0,IXMAX
                F(1,I) = OMEGA(I,L,1) + STEP*NONLI(I,L,1)
                F(2,I) = OMEGA(I,L,2) + STEP*NONLI(I,L,2)
40           CONTINUE
C
C   Solve the system of equations.
C
             I1 = 1
             I2 = I1 + 8*IXMAX + 8
             CALL TRIDIA(A,IXMAX+1,W,F,WORK(I1),WORK(I2))
C
C   Insert the solution in OMEGA.
C
             DO 45 I = 0,IXMAX
                OMEGA(I,L,1) = W(1,I)
                OMEGA(I,L,2) = W(2,I)
45           CONTINUE
C
C   End loop for each harmonic.
C
             COLD(L) = CNEW(L)
50        CONTINUE
C
C   Send possible print out.
C

          IF (SIGMA.GE.WAX) THEN
             CALL BEWRI2(SIGMA,OMEGA,IXMAX,NHARM)
             wax=wax+dwax
          END IF
C
C
          IF (BCOUNT.LE.NUMBE) THEN
             IF (WBEAM(BCOUNT).LE.SIGMA) THEN
                CALL BEWRI(WINDWX,SIGMA,OMEGA,IXMAX,NHARM)
                BCOUNT = BCOUNT + 1
             END IF
          END IF
          IF (SIGMA.GE.STOP) THEN
             GO TO 999
          END IF
C
C   End fully implicit loop.
C
100    CONTINUE
C
C   Continue with the Richtmyer procedure.
C
105    CONTINUE
       STEP = STEPP(SIGMA)
       SIG = SIGMA + STEP
       OLDSIG = SIGMA
       IF (BCOUNT.LE.NUMBE) THEN
          SIGMA = MIN(SIG,WBEAM(BCOUNT),STOP)
       ELSE
          SIGMA = STOP
       END IF
       STEP = SIGMA - OLDSIG
C
C   Compute the coefficients and the nonlinear
C   contributions.
C
       SIGC = SIGMA
       SIGK = OLDSIG + STEP/2
       CALL COEFF(SIGC,SIGK,CNEW,KHALF,NHARM)
       CALL NONLIN(SIGMA,OMEGA,IXMAX,NHARM,NONLI)
       R = STEP/XSTEP**2
C
C   Loop for each harmonic L.
C
       DO 150 L = 1,NHARM
C
C   Set up the appropriate systems of equations in the
C   way required for TRIDIA.
C
          DIAG = 1 + STEP*CNEW(L)/2
          OFFDIA = R*KHALF(L)
C
C   First the diagonal 2*2 sub-matrices.
C
          A(1,1,0,0) = DIAG
          A(1,2,0,0) = 2*OFFDIA
          A(2,1,0,0) = - A(1,2,0,0)
          A(2,2,0,0) = DIAG
          DO 110 I = 1,IXMAX
             A(1,1,0,I) = DIAG
             A(1,2,0,I) = OFFDIA
             A(2,1,0,I) = - OFFDIA
             A(2,2,0,I) = DIAG
110       CONTINUE
C
C   Then the sub-matrices above the diagonal.
C
          A(1,2,1,0) = - 2*OFFDIA
          A(2,1,1,0) = - A(1,2,1,0)
          DO 120 I = 1,IXMAX - 1
             A(1,2,1,I) = - OFFDIA* (1+1/DBLE(2*I))/2
             A(2,1,1,I) = - A(1,2,1,I)
120       CONTINUE
C
C   and the sub-matrices below the diagonal.
C
          DO 130 I = 1,IXMAX
             A(1,2, - 1,I) = - OFFDIA* (1-1/DBLE(2*I))/2
             A(2,1, - 1,I) = - A(1,2, - 1,I)
130       CONTINUE
C
C   Set up the right hand side.
C
          DIAG = 1 - STEP*COLD(L)/2
          OFFDIA = R*KHALF(L)
          F(1,0) = DIAG*OMEGA(0,L,1) + 2*OFFDIA*
     +            (OMEGA(1,L,2)-OMEGA(0,L,2)) + STEP*NONLI(0,L,1)
          F(2,0) = DIAG*OMEGA(0,L,2) - 2*OFFDIA*
     +            (OMEGA(1,L,1)-OMEGA(0,L,1)) + STEP*NONLI(0,L,2)
          DO 140 I = 1,IXMAX - 1
             FAKT1 = 1 + 1/DBLE(2*I)
             FAKT2 = 1 - 1/DBLE(2*I)
             F(1,I) = DIAG*OMEGA(I,L,1) +
     +                OFFDIA* (FAKT1*OMEGA(I+1,L,2)-2*OMEGA(I,L,2)
     +                +FAKT2*OMEGA(I-1,L,2))/2 + STEP*NONLI(I,L,1)
             F(2,I) = DIAG*OMEGA(I,L,2) -
     +                OFFDIA* (FAKT1*OMEGA(I+1,L,1)-2*OMEGA(I,L,1)
     +                +FAKT2*OMEGA(I-1,L,1))/2 + STEP*NONLI(I,L,2)
140       CONTINUE
```

```
          FAKT2 = 1 - 1/DBLE(2*IXMAX)
          F(1,IXMAX) = DIAG*OMEGA(IXMAX,L,1) +
     +                 OFFDIA* (FAKT2*OMEGA(IXMAX-1,L,2)-
     +                 2*OMEGA(IXMAX,L,2))/2
     +                 + STEP*NONLI(IXMAX,L,1)
          F(2,IXMAX) = DIAG*OMEGA(IXMAX,L,2) -
     +                 OFFDIA* (FAKT2*OMEGA(IXMAX-1,L,1)-
     +                 2*OMEGA(IXMAX,L,1))/2
     +                 + STEP*NONLI(IXMAX,L,2)
C
C   Solve the system of equations.
C
          I1 = 1
          I2 = I1 + 8*IXMAX + 8
          CALL TRIDIA(A,IXMAX+1,W,F,WORK(I1),WORK(I2))
C
C   Insert the solution in OMEGA.
C
          DO 145 I = 0,IXMAX
              OMEGA(I,L,1) = W(1,I)
              OMEGA(I,L,2) = W(2,I)
145       CONTINUE
C
C   End loop for each harmonic.
C
          COLD(L) = CNEW(L)
150   CONTINUE

C  **************************
C   Send possible print out.
C  **************************

          IF (SIGMA.GE.WAX) THEN
              CALL BEWRI2(SIGMA,OMEGA,IXMAX,NHARM)
              wax=wax+dwax
          END IF
C
      IF (BCOUNT.LE.NUMBE) THEN
          IF (WBEAM(BCOUNT).LE.SIGMA) THEN
              CALL BEWRI(WINDWX,SIGMA,OMEGA,IXMAX,NHARM)
              BCOUNT = BCOUNT + 1
          END IF
      END IF
C
      IF (SIGMA.GE.STOP) THEN
          GO TO 999
      END IF
C
C   End Richtmyer loop.
C
      GO TO 105
C
C***END KZKCYL
C
999   RETURN
      END
      SUBROUTINE TRIDIA(A,IXMAX,W,F,L,U)
C***BEGIN PROLOGUE TRIDIA
C***AUTHOR Jarle Berntsen, University of Bergen,
C           EDB-senteret, Herman Fossgt 6, 5007 Bergen, NORWAY.
C***DATE WRITTEN 880810
C***PURPOSE
C      BLOTRI SOLVES A BLOCK-THREE DIAGONAL SYSTEM OF EQUATIONS
C      BY BLOCK GAUSSIAN ELIMINATION.
C      40*IXMAX OPERATIONS ARE NEEDED.
C
C      THE BLOCKS ARE 2*2.
C      THE 2*2 SUB-MATRICES BELOW AND ABOVE THE MAIN DIAGONAL OF
C      A MUST HAVE ZEROS ON THEIR DIAGONALS.
C
C***DESCRIPTION
C
C      ON ENTRY
C             A       DOUBLE PRECISION(2,2,-1:1,IXMAX).
C                     DEFINES THE THREE-DIAGONAL MATRIX.
C                     THE FIRST TWO INDEXES DEFINES THE
C                     POSITION OF AN ELEMENT WITHIN A 2*2
C                     SUB-MATRIX. THE THIRD INDEX DEFINES THE
C                     DISTANCE OF THE SUB-MATRIX FROM THE
C                     DIAGONAL.
C                     THE LAST INDEX DEFINES THE ROW NUMBER
C                     OF THE SUB-MATRIX.
C                     THIS INDEX CONVENTION IS ALSO USED FOR
C                     L AND U. UNCHANGED ON EXIT.
C
C             IXMAX   INTEGER. THE NUMBER OF DIAGONAL
C                     SUB-MATRICES.
C
C             F       DOUBLE PRECISION(2,IXMAX). DEFINES THE
C                     RIGHT-HAND SIDE. MODIFIED ON EXIT.
C
C             L       DOUBLE PRECISION(2,2,-1:0,IXMAX).
C                     WORKING SPACE
C                     USED FOR THE SUBMATRICES ON AND BELOW
C                     THE DIAGONAL
C                     OF THE L-MATRIX OF THE LU-DECOMPOSITION.
C
C             U       DOUBLE PRECISION(2,2,1:1,IXMAX). WORKING
C                     SPACE USED FOR THE SUBMATRICES ABOVE
C                     THE DIAGONAL OF THE U MATRIX OF THE
C                     LU-DECOMP. THE DIAGONAL SUBMATRICES OF
C                     U ARE UNIT 2*2.
C
C       ON RETURN
C
C             W       DOUBLE PRECISION(2,IXMAX).
C                     CONTAINS THE SOLUTION.
C
C***ROUTINES CALLED NONE
C***END PROLOGUE TRIDIA
C
C   Global variables.
C
      INTEGER IXMAX
      DOUBLE PRECISION A(2,2, - 1:1,IXMAX)
      DOUBLE PRECISION W(2,IXMAX),F(2,IXMAX),L(2,2, - 1:0,IXMAX)
      DOUBLE PRECISION U(2,2,1:1,IXMAX)
C
C   Local variables.
C
      INTEGER I,J
      DOUBLE PRECISION F1,F2,L21,U22
C
C       LU-DECOMPOSE.
C
C       FIRST STEP. (FIRST ROW)
C
C       L(0,1)=A(0,1)
C
C***FIRST EXECUTABLE STATEMENT TRIDIA
C
      DO 1 I = 1,2
         DO 1 J = 1,2
1     L(I,J,0,1) = A(I,J,0,1)
C
C       THE 2*2 SYSTEMS ARE SOLVED AS FOLLOWS:(AX=F)
C       L21=A21/A11
C       U22=A22-L21*A12
C       F2=F2-L21*F1
C       X2=F2/U22
C       X1=(F1-A12*X2)/A11
C
C       L(0,1)*U(1,1)=A(1,1)
C
      L21 = L(2,1,0,1)/L(1,1,0,1)
      U22 = L(2,2,0,1) - L21*L(1,2,0,1)
C       F2=A(2,1,1,1)-L21*A(1,1,1,1)(WE EXPLOIT ALL ZEROS)
      F2 = A(2,1,1,1)
      U(2,1,1,1) = F2/U22
C       U(1,1,1,1)=(A(1,1,1,1)-L(1,2,0,1)*U(2,1,1,1))/L(1,1,0,1)
      U(1,1,1,1) = - L(1,2,0,1)*U(2,1,1,1)/L(1,1,0,1)
C
      F1 = A(1,2,1,1)
      F2 = - L21*F1
      U(2,2,1,1) = F2/U22
      U(1,2,1,1) = (F1-L(1,2,0,1)*U(2,2,1,1))/L(1,1,0,1)
C
C       SECOND AND GENERAL STEP.
C
      DO 10 I = 2,IXMAX - 1
C
C       L(-1,I)=A(-1,I)
C
C         L(1,1,-1,I)=0
C         L(2,2,-1,I)=0
          L(1,2, - 1,I) = A(1,2, - 1,I)
          L(2,1, - 1,I) = A(2,1, - 1,I)
C
C       L(0,I)=A(0,I)-L(-1,I)*U(1,I-1)
C
          L(1,1,0,I) = A(1,1,0,I) - L(1,2, - 1,I)*U(2,1,1,I-1)
          L(1,2,0,I) = A(1,2,0,I) - L(1,2, - 1,I)*U(2,2,1,I-1)
          L(2,1,0,I) = A(2,1,0,I) - L(2,1, - 1,I)*U(1,1,1,I-1)
          L(2,2,0,I) = A(2,2,0,I) - L(2,1, - 1,I)*U(1,2,1,I-1)
C
C       L(0,I)*U(1,I)=A(1,I)
```

```
C
         L21 = L(2,1,0,I)/L(1,1,0,I)
         U22 = L(2,2,0,I) - L21*L(1,2,0,I)
         F2 = A(2,1,1,I)
         U(2,1,1,I) = F2/U22
         U(1,1,1,I) = - L(1,2,0,I)*U(2,1,1,I)/L(1,1,0,I)
         F1 = A(1,2,1,I)
         F2 = - L21*F1
         U(2,2,1,I) = F2/U22
         U(1,2,1,I) = (F1-L(1,2,0,I)*U(2,2,1,I))/L(1,1,0,I)
10    CONTINUE
C
C       END MAIN LOOP IN LU-DECOMP.
C
C       THIRD AND LAST STEP.(LAST ROW)
C
C       L(-1,IXMAX)=A(-1,IXMAX)
C
C       L(1,1,-1,IXMAX)=0
C       L(2,2,-1,IXMAX)=0
      L(1,2, - 1,IXMAX) = A(1,2, - 1,IXMAX)
      L(2,1, - 1,IXMAX) = A(2,1, - 1,IXMAX)
C
C       L(0,IXMAX)=A(0,IXMAX)-L(-1,IXMAX)*U(1,IXMAX-1)
C
      L(1,1,0,IXMAX) = A(1,1,0,IXMAX) -

     +                 L(1,2, - 1,IXMAX)*U(2,1,1,IXMAX-1)
      L(1,2,0,IXMAX) = A(1,2,0,IXMAX) -
     +                 L(1,2, - 1,IXMAX)*U(2,2,1,IXMAX-1)
      L(2,1,0,IXMAX) = A(2,1,0,IXMAX) -
     +                 L(2,1, - 1,IXMAX)*U(1,1,1,IXMAX-1)
      L(2,2,0,IXMAX) = A(2,2,0,IXMAX) -
     +                 L(2,1, - 1,IXMAX)*U(1,2,1,IXMAX-1)
C
C       END LU FACTORISATION.
C
C       FORWARD SUBSTITUTION.
C
C       FIRST STEP.
C
      L21 = L(2,1,0,1)/L(1,1,0,1)
      U22 = L(2,2,0,1) - L21*L(1,2,0,1)
      F(2,1) = F(2,1) - L21*F(1,1)
      F(2,1) = F(2,1)/U22
      F(1,1) = (F(1,1)-L(1,2,0,1)*F(2,1))/L(1,1,0,1)
C
C       SECOND AND GENERAL STEP.
C
      DO 20 I = 2,IXMAX
         L21 = L(2,1,0,I)/L(1,1,0,I)
         U22 = L(2,2,0,I) - L21*L(1,2,0,I)
         F1 = F(1,I) - L(1,2, - 1,I)*F(2,I-1)
         F2 = F(2,I) - L(2,1, - 1,I)*F(1,I-1)
         F2 = F2 - L21*F1
         F(2,I) = F2/U22
         F(1,I) = (F1-L(1,2,0,I)*F(2,I))/L(1,1,0,I)
20    CONTINUE
C
C       END FORWARD SUBSTITUTION.
C
C       BACK SUBSTITUTION.
C
C       FIRST STEP.
C
      W(1,IXMAX) = F(1,IXMAX)
      W(2,IXMAX) = F(2,IXMAX)
C
C       SECOND AND GENERAL STEP.
C
      DO 30 I = IXMAX - 1,1, - 1
         W(1,I) = F(1,I) - U(1,1,1,I)*W(1,I+1)
     +              - U(1,2,1,I)*W(2,I+1)
         W(2,I) = F(2,I) - U(2,1,1,I)*W(1,I+1)
     +              - U(2,2,1,I)*W(2,I+1)
30    CONTINUE
C
C       END BACK SUBSTITUTION.
C
C***END TRIDIA
C
      RETURN
      END
```

tbe_maalt.f

Versjon av Bergenkoden som henter måledata til bruk som startbetingelse. Filene `adata`, `bdata` og `ydata` skal være ASCII-filer som inneholder et enkelt tall på hver linje. De skal inneholde de normaliserte Fourierkoeffisientene a_1 og b_1, og transversal posisjon normalisert til kildens radius. Programmet kan lett utvides til å ta med høyere harmoniske i kildebetingelsen.

```
C
C     BEAM --- Bergen code
C     Transformed beam equation
C
C
      PROGRAM BEAM
      INTEGER IXMAX,NHARM,NUMBE,t,antall
      PARAMETER (IXMAX=5000,NHARM=40,NUMBE=1)
      DOUBLE PRECISION OMEGA(0:IXMAX,NHARM,2),WINDWX,START,STOP,
     +                 WBEAM(NUMBE),COLD(NHARM),CNEW(NHARM),
     +                 KHALF(NHARM),NONLI(0:IXMAX,NHARM,2),
     +                 A(2,2, - 1:1,0:IXMAX),W(2,0:IXMAX),
     +                 F(2,0:IXMAX),WORK(12*IXMAX+12),
     +                 FREQ,RADIUS,SSPEED,ALFA2,BETA,MACH,PI,
     +                 r0,rd,k,d
      EXTERNAL COEFF,NONLIN,STEPP,BEWRI
      DOUBLE PRECISION PABSOR,PNONLI,ISSTEP
      CHARACTER*10 transv,ax,desc,fn
      COMMON /PARAM/PABSOR,PNONLI
      COMMON /AXSTEP/ISSTEP

      fn = 'innfil' ! Name for the parameter file
      antall = 3   ! Number of lines in the parameter file
      OPEN(UNIT=9,FILE=fn,STATUS="unknown")
C     Først tre filnavn, så parametere
        write(9,2) 'tr301','ax301','dsc301',
     +             'M = ',0.7d-4,'a = ',27.d-3,
     +             'beta = ',3.5
        write(9,2) 'tr302','ax302','dsc302',
     +             'M = ',0.8d-4,'a = ',27.d-3,
     +             'beta = ',3.5
        write(9,2) 'tr303','ax303','dsc303',
     +             'M = ',0.9d-4,'a = ',27.d-3,
     +             'beta = ',3.5
      CLOSE(9)

2     FORMAT(a5,1x,a5,1x,a6,2x,a4,e16.10,2x,a4,e16.10,2x,
     +       a7,e16.10)
3     FORMAT(a5,1x,a5,1x,a6,6x,e16.10,6x,e16.10,9x,e16.10)
C
C     Input fra iterasjon.m
c3     format(a4,7x,e16.10,7x,e16.10,10x,e16.10)
C
C     Run the program for each line of fn
C
      OPEN(UNIT=9,FILE=fn,STATUS='old')
      DO 5 t=1,antall
READ(9,3) transv,ax,desc,mach,radius,beta
        OPEN (UNIT=10,FILE=transv,STATUS='unknown')
        OPEN (UNIT=11,FILE=ax,STATUS='unknown')
        OPEN (unit=12,file=desc,STATUS='unknown')
C
C       Read input (For beregning av PABSOR og PNONLI)
C
C       Frekvens  Hz
        FREQ = 1.1d6
C       Radius, m (X = x/RADIUS)
c        RADIUS = 25.d-3
C       Lydhastighet, m/s
        sspeed = 1490.0
C       Absorbsjonskonstant, s^2/m
        alfa2 = 2.5d-14
C       Ikkelinearitetsparameter
c        beta = 3.5
C       Machtall
c        mach = 1.0D-4
C       Bredde paa sinus-avrunding av kilden
        d = 0.
```

```
C
        PI = 4*ATAN(1.0D0)
        k  = 2*pi*freq/sspeed
        r0 = radius**2*k/2
        rd = 1/(beta*mach*k)
        PABSOR = alfa2*freq**2*r0
        PNONLI = r0/rd
c
c
        WINDWX = 60.
c        ISSTEP = 3.5D-3
        isstep = 2.d-3
        wbeam(1) = 0.75
C
C       Initiate the Fourier coefficients.
C
        CALL INIT(WINDWX,OMEGA,IXMAX,NHARM,d)
C
C       Integrate the differential equations.
C
        START = 0
        STOP = WBEAM(NUMBE)
C
        WRITE(12,*) 'c = ',sspeed,' a = ',radius,' f = ',freq
        WRITE(12,*) 'pabsorb = ',pabsor,' pnonlin = ',pnonli
        WRITE(12,*) 'Machtall = ',mach,' Rayl-dist = ',r0,
     +  ' Sjokkavst = ',rd
        WRITE(12,*) 'Snittposisjoner: ',(wbeam(i),i=1,numbe)
        WRITE(12,*) 'Antall harmoniske: ',nharm,'d = ',d
        CLOSE(12)
C
        WRITE(6,*) 'kzkcyl, M =',mach,', d = ',d,', nharm = ',
     +            nharm
        CALL KZKCYL(OMEGA,IXMAX,NHARM,WINDWX,START,STOP,WBEAM,
     +             NUMBE,COEFF,NONLIN,STEPP,BEWRI,COLD,CNEW,
     +             KHALF,NONLI,A,W,F,WORK)
      CLOSE(10)
      CLOSE(11)
      CLOSE(12)
5     CONTINUE
      CLOSE(9)
      STOP
      END

      SUBROUTINE INIT(WINDWX,OMEGA,IXMAX,NHARM,d)
      INTEGER I,J,IXMAX,NHARM,n
      DOUBLE PRECISION OMEGA(0:IXMAX,NHARM,2),WINDWX,d,y(61),
     +                  X(0:ixmax),dm(2),a0(61,2),b0(61,2),
     +                  a(0:ixmax),b(0:ixmax),dm2(0:ixmax)
      DOUBLE PRECISION USTEP,FA
C
C     Retrieve measured data for the boundary condition
C
      open(unit=15,file='adata',status='unknown')
      open(unit=16,file='bdata',status='unknown')
      open(unit=17,file='ydata',status='unknown')
6     format(e16.10)
      do 7 i = 1,61
read(15,6) a0(i,1)
read(16,6) b0(i,1)
read(17,6) y(i)
a0(i,2) = a0(i,1)
b0(i,2) = b0(i,1)
7     continue
      close(15)
      close(16)
      close(17)
      dm(1) = 1  !       array needed
      dm(2) = 2  !       in itplbv
      DO 20 I = 0,IXMAX
          IF (NHARM.GT.1) THEN
              DO 10 J = 2,NHARM
                  OMEGA(I,J,1) = 0
                  OMEGA(I,J,2) = 0
10            CONTINUE
          END IF
  dm2(i) = 1  !  array needed in itplbv
          X(i) = I* WINDWX/DBLE(IXMAX+1)
20    CONTINUE
C
C     Use interpolation to insert the boundary condition
C     into the model
C
      n = ixmax + 1
      call itplbv(12,61,2,y,dm,a0,n,x,dm2,a)
      call itplbv(12,61,2,y,dm,b0,n,x,dm2,b)
      do 25 i = 0,ixmax
          omega(i,1,1) = ustep(x(i),d)*a(i);
```

```
  omega(i,1,2) = ustep(x(i),d)*b(i);
25    continue


      RETURN
      END
C
C     p(U,t) = A cos(wt) + B sin(wt)
C            = USTEP(X)*sin(wt + FA(X))
C
      DOUBLE PRECISION FUNCTION FA(U)
      DOUBLE PRECISION U
      FA = (ABS(U))**2
C      FA = 0  ! Uniform stempelkilde?
      RETURN
      END
      DOUBLE PRECISION FUNCTION USTEP(U,d)
      DOUBLE PRECISION U,d,pi
      IF (abs(U).le.(1.5d0 - d/2)) THEN
c      IF (abs(U).le.(1.d0 - d/2)) THEN
         USTEP = 1
      ELSE
          if (abs(U).le.(1.d0 + d/2)) then
              pi = 4*atan(1.d0)
              USTEP = 0.5*(1 + sin(pi/d*(1 - abs(U))))
          else
              USTEP = 0
          end if
      END IF
      RETURN
      END
      SUBROUTINE BEWRI(WINDWX,SIGMA,OMEGA,IXMAX,NHARM)
      INTEGER NHARM,IXMAX,J,I
      DOUBLE PRECISION SIGMA,OMEGA(0:IXMAX,NHARM,2),WINDWX
      DOUBLE PRECISION S,X,AMP,AMPL,DB,AMPAX,amp2,db2,ampax3
      DOUBLE PRECISION amp3,amp4,amp5
      S = (1.DO+SIGMA)
C      write(10,*) sngl(sigma)
      DO 20 J = 1,1
C         AMPAX = AMPL(OMEGA(0,J,1),OMEGA(0,J,2),S)
C         AMPax3 = AMPL(OMEGA(0,2,1),OMEGA(0,2,2),S)
          DO 10 I = 0,IXMAX
              X = I*WINDWX/DBLE(IXMAX+1)
              AMP = AMPL(OMEGA(I,J,1),OMEGA(I,J,2),S)
C             IF (AMP.NE.0) THEN
C                DB = AMP/AMPAX
C                DB = -20*LOG10(DB)
C                if(db.gt.220)then
C                   db = 0
C                endif
C             ELSE
C                DB = 0
C             END IF
              AMP2 = AMPL(OMEGA(I,2,1),OMEGA(I,2,2),S)
C             IF (AMP2.NE.0) THEN
C                DB2 = AMP2/AMPax3
C                DB2 = -20*LOG10(DB2)
C                if (db2.gt.180) db2 = 0
C             ELSE
C                DB2 = 0
C             END IF
              AMP3 = AMPL(OMEGA(I,3,1),OMEGA(I,3,2),S)
              AMP4 = AMPL(OMEGA(I,4,1),OMEGA(I,4,2),S)
              AMP5 = AMPL(OMEGA(I,5,1),OMEGA(I,5,2),S)
            WRITE (10,22) SNGL(X),SNGL(amp),SNGL(amp2),
     +                    SNGL(amp3),SNGL(amp4),SNGL(amp5)
10        CONTINUE
20    CONTINUE
22    FORMAT(6(e16.10,1x))
      WRITE(6,*) 'transv updated'
      RETURN
      END
C
C     On the axis:
C
      SUBROUTINE BEWRI2(SIGMA,OMEGA,IXMAX,NHARM)
      DOUBLE PRECISION SIGMA,OMEGA(0:IXMAX,NHARM,2)
      DOUBLE PRECISION AMP(5),AMPL,S
      INTEGER NHARM,IXMAX,I
      S = (1.DO+SIGMA)
      DO 25 I = 1,5
          AMP(I) = AMPL(OMEGA(0,I,1),OMEGA(0,I,2),S)
25    CONTINUE
      WRITE(11,28) sngl(SIGMA),(AMP(I),I=1,5)
28    FORMAT(6(e16.10,1x))
      RETURN
      END
      SUBROUTINE NONLIN(SIGMA,OMEGA,IXMAX,NHARM,NONLI)
```

```
      INTEGER NHARM,IXMAX,I,J,N
      DOUBLE PRECISION OMEGA(0:IXMAX,NHARM,2)
      DOUBLE PRECISION NONLI(0:IXMAX,NHARM,2)
      DOUBLE PRECISION SIGMA,FAKT,S,PABSOR,PNONLI
      COMMON /PARAM/PABSOR,PNONLI
      S = (1.D0+SIGMA)
      DO 100 N = 1,NHARM
          DO 29 I = 0,IXMAX
              NONLI(I,N,1) = 0
              NONLI(I,N,2) = 0
29        CONTINUE
          DO 50 I = 0,IXMAX
              DO 30 J = 1,N - 1
                  NONLI(I,N,1) = NONLI(I,N,1) +
     +                           (OMEGA(I,J,2)*OMEGA(I,N-J,1))
                  NONLI(I,N,2) = NONLI(I,N,2) +
     +                           OMEGA(I,J,2)*OMEGA(I,N-J,2) -
     +                           OMEGA(I,J,1)*OMEGA(I,N-J,1)
30            CONTINUE
              NONLI(I,N,2) = 0.5*NONLI(I,N,2)
              DO 40 J = N + 1,NHARM
                  NONLI(I,N,1) = NONLI(I,N,1) +
     +                           OMEGA(I,J-N,1)*OMEGA(I,J,2) -
     +                           OMEGA(I,J-N,2)*OMEGA(I,J,1)
                  NONLI(I,N,2) = NONLI(I,N,2) -
     +                           OMEGA(I,J-N,1)*OMEGA(I,J,1) -
     +                           OMEGA(I,J-N,2)*OMEGA(I,J,2)
40            CONTINUE
50        CONTINUE
          FAKT = PNONLI*N/ (2*S)
          DO 60 I = 0,IXMAX
              NONLI(I,N,1) = FAKT*NONLI(I,N,1)
              NONLI(I,N,2) = FAKT*NONLI(I,N,2)
60        CONTINUE
100   CONTINUE
      RETURN
      END
      DOUBLE PRECISION FUNCTION AMPL(FA,FB,S)
C
C     FUNCTION CALCULATING AMPLITUDE.
C
      DOUBLE PRECISION FA,FB,S,SD
      SD = FA**2 + FB**2
      IF (S.NE.0) THEN
          AMPL = DSQRT(SD)/S
      ELSE
          AMPL = DSQRT(SD)
      END IF
      END
      SUBROUTINE COEFF(SIGC,SIGK,C,K,NHARM)
      INTEGER NHARM,J
      DOUBLE PRECISION SIGC,SIGK,C(NHARM),K(NHARM)
      DOUBLE PRECISION PABSOR,PNONLI,SC,SK
      COMMON /PARAM/PABSOR,PNONLI
      SC = (1+SIGC)
      SK = (1+SIGK)
      DO 10 J = 1,NHARM
          C(J) = PABSOR* (J**2)
          K(J) = - 1.0/ (4*J* (SK**2))
10    CONTINUE
      RETURN
      END
      DOUBLE PRECISION FUNCTION STEPP(SIGMA)
      DOUBLE PRECISION SIGMA,ISSTEP
      COMMON /AXSTEP/ISSTEP
      STEPP = ISSTEP*(1 + SIGMA)**2
      RETURN
      END
      SUBROUTINE KZKCYL(OMEGA,IXMAX,NHARM,WINDWX,START,STOP,WBEAM,
     +                  NUMBE,COEFF,NONLIN,STEPP,BEWRI,COLD,CNEW,
     +                  KHALF,NONLI,A,W,F,WORK)
C***BEGIN PROLOGUE KZKCYL
C***AUTHOR Jarle Berntsen, University of Bergen,
C          EDB-senteret, Herman Fossgt 6, 5007 Bergen, NORWAY.
C***PURPOSE KZKCYL computes approximations to the Khokhlov-
C           Zabolotzkaya-Kuznetsov equation,see [1,2].
C           After a Fourier-series expansion of the pressure,
C           we may compute approximations to the KZK equation by
C           solving a system of nonlinear parabolic equations of
C           the form:
C
C       dg(n)/dsigma = -c(n,sigma)g(n)
C                      + k(n,sigma)Laplacian(h(n))
C                      + il1(n,sigma,g,h)
C
C       dh(n)/dsigma = -c(n,sigma)h(n)
C                      - k(n,sigma)Laplacian(g(n))
C                      + il2(n,sigma,g,h)
C
C           The terms on the right hand side of the equality sign
C           are due to (from left to right) absorbtion,
C           diffraction and nonlinearity.
C           In KZKCYL the Laplacian is cylindrical coordinates.
C
C***DESCRIPTION Two numerical techniques are applied in KZKCYL in
C               order to compute approximations to the system of
C               equations given above.
C               For uniform sources which are of particular
C               interest, Gibbs oscillations appear in the
C               numerical solution.
C               In order to damp these we first take a few steps
C               with a fully implicit method.
C               In order to control the damping
C               the S first harmonics of the Fourier expansion
C               of the initial values are kept to 99.9 per cent
C               of their full strength.
C               S is set as a parameter in KZKCYL.
C
C               After having removed the Gibbs oscillations we
C               continue with the Trapezoidal rule
C               (or Crank-Nicolson method or on this particular
C               problem, Richtmyers procedure)
C               which has 2'nd order accuracy and is
C               unconditionally stable.
C
C               The linear systems of equations that have to be
C               solved in each step are solved with the routine
C               TRIDIA.
C
C   ON ENTRY
C
C     OMEGA  Real array of dimension (0:IXMAX,NHARM,2).
C            Contains in each step approximative values to g(n)
C            and h(n).
C            The first index defines the values of x for which
C            g(n) and h(n) are approximated.
C            The second index defines the harmonic number.
C            If the last index is 1, approximations to g(n) are
C            given. If the last index is 2, approximations to
C            h(n) are given.
C            On entry OMEGA must specify the initial values of
C            g(n) and h(n).
C
C     IXMAX  Integer.
C            Defines the number of interior grid points.
C
C     NHARM  Integer.
C            Defines the number of harmonics retained in the
C            numerical solution.
C
C     WINDWX Real.
C            WINDWX is the upper limit of x.
C
C     START  Real.
C            The starting value of sigma.
C
C     STOP   Real.
C            The final value of sigma.
C
C     WBEAM  Real array of dimension NUMBE.
C            Specifies the values of sigma where we want print
C            out of OMEGA.
C
C     NUMBE  Integer.
C            The length of the array WBEAM.
C
C     COEFF  Subroutine, supplied by the user.
C            COEFF must be declared as external in the calling
C            program. COEFF is called by KZKCYL to compute the
C            coefficients c(n,sigma) and k(n,sigma).
C
C            Its specification is:
C            SUBROUTINE COEFF(SIGC,SIGK,C,K,NHARM)
C            INTEGER NHARM
C            DOUBLE PRECISION SIGC,SIGK,C(NHARM),K(NHARM)
C
C            SIGC defines on entry the value of sigma for which
C            you want to compute the coefficients c(n,sigma).
C
C            SIGK defines on entry the value of sigma for which
C            you want to compute the coefficients k(n,sigma).
C
C            C contains on exit the
C            values of c(n,SIGC), n=1,...,NHARM.
C
C            K contains on exit the
```

```
C           values of k(n,SIGK), n=1,...,NHARM.
C
C     NONLIN Subroutine, supplied by the user.
C           NONLIN must be declared as external in the calling
C           program. NONLIN is called by KZKCYL to compute the
C           nonlinear terms
C           il1(n,sigma,g,h) and il2(n,sigma,g,h).
C
C           Its specification is:
C           SUBROUTINE NONLIN(SIGMA,OMEGA,IXMAX,NHARM,NONLI)
C           INTEGER IXMAX,NHARM
C           DOUBLE PRECISION SIGMA
C           DOUBLE PRECISION OMEGA(0:IXMAX,NHARM,2)
C           DOUBLE PRECISION NONLI(0:IXMAX,NHARM,2)
C
C           SIGMA defines on entry the value of sigma for which
C           you want to compute the nonlinear terms.
C
C           OMEGA defines on entry the computed approximations
C           to g(n) and h(n).
C
C           NONLI must on exit contain the nonlinear terms.
C           If the last index is 1, approx. to il1 is indicated.
C           If the last index is 2, approx. to il2 is indicated.
C
C     STEPP  Real function, supplied by the user.
C           STEPP must be declared as external in the calling
C           program. STEPP defines the step size in sigma
C           direction.
C
C
C           Its specification is:
C           DOUBLE PRECISION FUNCTION STEPP(SIGMA)
C           DOUBLE PRECISION SIGMA
C
C     BEWRI  Subroutine, supplied by the user.
C           BEWRI must be declared as external in the calling
C           program. BEWRI is called by KZKCYL to write
C           information in OMEGA to file(s).
C
C           Its specification is:
C           SUBROUTINE BEWRI(WINDWX,SIGMA,OMEGA,IXMAX,NHARM)
C           INTEGER IXMAX,NHARM
C           DOUBLE PRECISION WINDWX,SIGMA
C           DOUBLE PRECISION OMEGA(0:IXMAX,NHARM,2)
C
C  ON RETURN
C
C     OMEGA  contains the approximations to g(n) and h(n)
C           at sigma=STOP.
C
C     COLD   Real work array of dimension NHARM.
C
C     CNEW   Real work array of dimension NHARM.
C
C     KHALF  Real work array of dimension NHARM.
C
C     NONLI  Real work array of dimension (0:IXMAX,NHARM,2).
C
C     A      Real work array of dimension (2,2,-1:1,0:IXMAX).
C
C     W      Real work array of dimension (2,0:IXMAX).
C
C     F      Real work array of dimension (2,0:IXMAX).
C
C     WORK   Real work array of dimension (12*IXMAX+12).
C
C***REFERENCES
C     [1]Zabolotskaya,E.A. and Khokhlov,V.Pa,
C        Quasi-plane waves in the nonlinear acoustics of confined
C        beams, Sov.Phys.Acoust.,15,1969,pp.35-40.
C     [2]Kuznetsov,V.P.,Equations of nonlinear acoustics,
C        Sov.Phys.Acoust.,16,1971,pp.467-470.
C     [3]Hamilton,M.F. and Naze Tjotta,J. and Tjotta.S,
C        Nonlinear effects in the farfield of a directive sound
C        source, J.Acoust.Soc.Am.,78,1985,pp.202-216.
C     [4]Berntsen,J.,The numerical calculation of the Khokhlov-
C        Zabolotskaya-Kuznetsov equation for piston sources,
C        Unpublished paper,1988.
C***ROUTINES CALLED COEFF,NONLIN,STEPP,BEWRI,TRIDIA
C***END PROLOGUE KZKCYL
C
C   Global variables.
C
      INTEGER IXMAX,NHARM,NUMBE
      DOUBLE PRECISION OMEGA(0:IXMAX,NHARM,2),WINDWX,START,STOP,
     +                 WBEAM(NUMBE),STEPP,COLD(NHARM),
     +                 CNEW(NHARM),
```

```
     +                 KHALF(NHARM),NONLI(0:IXMAX,NHARM,2),
     +                 A(2,2, - 1:1,0:IXMAX),W(2,0:IXMAX),
     +                 F(2,0:IXMAX),
     +                 WORK(12*IXMAX+12)
      EXTERNAL COEFF,NONLIN,STEPP,BEWRI
C
C   Local variables.
C
      INTEGER L,I,BCOUNT,S,IMPSTP,IS,I1,I2
      DOUBLE PRECISION STEP,SIG,OLDSIG,SIGMA,R,DIAG,OFFDIA,PI,
     +                 XSTEP
      DOUBLE PRECISION SIGC,SIGK,FAKT1,FAKT2
      REAL wax,dwax
C
C   IMPSTP is the number of initial fully implicit steps.
C   The strength of the S first harmonics in the Fourier
C   expansion of the initial values must not be damped more than
C   0.1 percent when
C   taking the IMPSTP initial steps.
C
      PARAMETER (IMPSTP=10)
      PARAMETER (S=20)
C
C***FIRST EXECUTABLE STATEMENT KZKCYL
C
      PI = 4*ATAN(1.d0)
      BCOUNT = 1
      SIGMA = START
      dwax = 0.001
      wax = START + dwax
C
C   By setting START = STOP, we may print out the initial values.
C
      IF (START.EQ.STOP) THEN
          CALL BEWRI(WINDWX,SIGMA,OMEGA,IXMAX,NHARM)
          GO TO 999
      END IF
      XSTEP = ABS(WINDWX)/DBLE(IXMAX+1)
      SIGC = SIGMA
      SIGK = SIGMA
      CALL COEFF(SIGC,SIGK,COLD,KHALF,NHARM)
C
C   Go IMPSTP steps with the fully implicit method.
C
      DO 100 IS = 1,IMPSTP
C
C   Choose the step size such that the S first eigenvectors are
C   kept to 99.9 per cent of their initial strength after IMPSTP
C   steps.
          STEP = SQRT(2./ (1000.*IMPSTP))* (2*WINDWX)**2/
     +           (ABS(KHALF(1))*S**2*PI**2)

          step = 2.5d-3   ! 270202

    SIG = SIGMA + STEP
          OLDSIG = SIGMA
          IF (BCOUNT.LE.NUMBE) THEN
              SIGMA = MIN(SIG,WBEAM(BCOUNT),STOP)
          ELSE
              SIGMA = STOP
          END IF
          STEP = SIGMA - OLDSIG
C
C   Compute the coefficients and the nonlinear
C   contributions.
C
          SIGC = SIGMA
          SIGK = SIGMA
          CALL COEFF(SIGC,SIGK,CNEW,KHALF,NHARM)
          CALL NONLIN(SIGMA,OMEGA,IXMAX,NHARM,NONLI)
          R = STEP/XSTEP**2
C
C   Loop for each harmonic L.
C
          DO 50 L = 1,NHARM
C
C   Set up the appropriate systems of equations in the
C   way required for TRIDIA.
C
              DIAG = 1 + STEP*CNEW(L)
              OFFDIA = 2*R*KHALF(L)
C
C   First the diagonal 2*2 sub-matrices.
C
              A(1,1,0,0) = DIAG
              A(1,2,0,0) = 2*OFFDIA
              A(2,1,0,0) = - A(1,2,0,0)
              A(2,2,0,0) = DIAG
```

```
            DO 10 I = 1,IXMAX
               A(1,1,0,I) = DIAG
               A(1,2,0,I) = OFFDIA
               A(2,1,0,I) = - OFFDIA
               A(2,2,0,I) = DIAG
10          CONTINUE
C
C   Then the sub-matrices above the diagonal.
C
            A(1,2,1,0) = - 2*OFFDIA
            A(2,1,1,0) = - A(1,2,1,0)
            DO 20 I = 1,IXMAX - 1
               A(1,2,1,I) = - OFFDIA* (1+1/DBLE(2*I))/2
               A(2,1,1,I) = - A(1,2,1,I)
20          CONTINUE
C
C   and the sub-matrices below the diagonal.
C
            DO 30 I = 1,IXMAX
               A(1,2, - 1,I) = - OFFDIA* (1-1/DBLE(2*I))/2
               A(2,1, - 1,I) = - A(1,2, - 1,I)
30          CONTINUE
C
C   Define the right hand side.
C
            DO 40 I = 0,IXMAX
               F(1,I) = OMEGA(I,L,1) + STEP*NONLI(I,L,1)
               F(2,I) = OMEGA(I,L,2) + STEP*NONLI(I,L,2)
40          CONTINUE
C
C   Solve the system of equations.
C
            I1 = 1
            I2 = I1 + 8*IXMAX + 8
            CALL TRIDIA(A,IXMAX+1,W,F,WORK(I1),WORK(I2))
C
C   Insert the solution in OMEGA.
C
            DO 45 I = 0,IXMAX
               OMEGA(I,L,1) = W(1,I)
               OMEGA(I,L,2) = W(2,I)
45          CONTINUE
C
C   End loop for each harmonic.
C
            COLD(L) = CNEW(L)
50       CONTINUE
C
C   Send possible print out.
C

         IF (SIGMA.GE.WAX) THEN
            CALL BEWRI2(SIGMA,OMEGA,IXMAX,NHARM)
            wax=wax+dwax
         END IF
C
C
         IF (BCOUNT.LE.NUMBE) THEN
            IF (WBEAM(BCOUNT).LE.SIGMA) THEN
               CALL BEWRI(WINDWX,SIGMA,OMEGA,IXMAX,NHARM)
               BCOUNT = BCOUNT + 1
            END IF
         END IF
         IF (SIGMA.GE.STOP) THEN
            GO TO 999
         END IF
C
C   End fully implicit loop.
C
100   CONTINUE
C
C   Continue with the Richtmyer procedure.
C
105   CONTINUE
      STEP = STEPP(SIGMA)
      SIG = SIGMA + STEP
      OLDSIG = SIGMA
      IF (BCOUNT.LE.NUMBE) THEN
         SIGMA = MIN(SIG,WBEAM(BCOUNT),STOP)
      ELSE
         SIGMA = STOP
      END IF
      STEP = SIGMA - OLDSIG
C
C   Compute the coefficients and the nonlinear
C   contributions.
C
      SIGC = SIGMA
      SIGK = OLDSIG + STEP/2
      CALL COEFF(SIGC,SIGK,CNEW,KHALF,NHARM)
      CALL NONLIN(SIGMA,OMEGA,IXMAX,NHARM,NONLI)
      R = STEP/XSTEP**2
C
C   Loop for each harmonic L.
C
      DO 150 L = 1,NHARM
C
C   Set up the appropriate systems of equations in the
C   way required for TRIDIA.
C
         DIAG = 1 + STEP*CNEW(L)/2
         OFFDIA = R*KHALF(L)
C
C   First the diagonal 2*2 sub-matrices.
C
         A(1,1,0,0) = DIAG
         A(1,2,0,0) = 2*OFFDIA
         A(2,1,0,0) = - A(1,2,0,0)
         A(2,2,0,0) = DIAG
         DO 110 I = 1,IXMAX
            A(1,1,0,I) = DIAG
            A(1,2,0,I) = OFFDIA
            A(2,1,0,I) = - OFFDIA
            A(2,2,0,I) = DIAG
110      CONTINUE
C
C   Then the sub-matrices above the diagonal.
C
         A(1,2,1,0) = - 2*OFFDIA
         A(2,1,1,0) = - A(1,2,1,0)
         DO 120 I = 1,IXMAX - 1
            A(1,2,1,I) = - OFFDIA* (1+1/DBLE(2*I))/2
            A(2,1,1,I) = - A(1,2,1,I)
120      CONTINUE
C
C   and the sub-matrices below the diagonal.
C
         DO 130 I = 1,IXMAX
            A(1,2, - 1,I) = - OFFDIA* (1-1/DBLE(2*I))/2
            A(2,1, - 1,I) = - A(1,2, - 1,I)
130      CONTINUE
C
C   Set up the right hand side.
C
         DIAG = 1 - STEP*COLD(L)/2
         OFFDIA = R*KHALF(L)
         F(1,0) = DIAG*OMEGA(0,L,1) + 2*OFFDIA*
     +            (OMEGA(1,L,2)-OMEGA(0,L,2)) + STEP*NONLI(0,L,1)
         F(2,0) = DIAG*OMEGA(0,L,2) - 2*OFFDIA*
     +            (OMEGA(1,L,1)-OMEGA(0,L,1)) + STEP*NONLI(0,L,2)
         DO 140 I = 1,IXMAX - 1
            FAKT1 = 1 + 1/DBLE(2*I)
            FAKT2 = 1 - 1/DBLE(2*I)
            F(1,I) = DIAG*OMEGA(I,L,1) +
     +               OFFDIA* (FAKT1*OMEGA(I+1,L,2)-2*OMEGA(I,L,2)
     +               +FAKT2*OMEGA(I-1,L,2))/2 + STEP*NONLI(I,L,1)
            F(2,I) = DIAG*OMEGA(I,L,2) -
     +               OFFDIA* (FAKT1*OMEGA(I+1,L,1)-2*OMEGA(I,L,1)
     +               +FAKT2*OMEGA(I-1,L,1))/2 + STEP*NONLI(I,L,2)
140      CONTINUE
         FAKT2 = 1 - 1/DBLE(2*IXMAX)
         F(1,IXMAX) = DIAG*OMEGA(IXMAX,L,1) +
     +                OFFDIA* (FAKT2*OMEGA(IXMAX-1,L,2)-
     +                2*OMEGA(IXMAX,L,2))/2
     +                + STEP*NONLI(IXMAX,L,1)
         F(2,IXMAX) = DIAG*OMEGA(IXMAX,L,2) -
     +                OFFDIA* (FAKT2*OMEGA(IXMAX-1,L,1)-
     +                2*OMEGA(IXMAX,L,1))/2
     +                + STEP*NONLI(IXMAX,L,2)
C
C   Solve the system of equations.
C
         I1 = 1
         I2 = I1 + 8*IXMAX + 8
         CALL TRIDIA(A,IXMAX+1,W,F,WORK(I1),WORK(I2))
C
C   Insert the solution in OMEGA.
C
         DO 145 I = 0,IXMAX
            OMEGA(I,L,1) = W(1,I)
            OMEGA(I,L,2) = W(2,I)
145      CONTINUE
C
C   End loop for each harmonic.
C
         COLD(L) = CNEW(L)
```

```
150   CONTINUE

C  **************************
C   Send possible print out.
C  **************************

          IF (SIGMA.GE.WAX) THEN
              CALL BEWRI2(SIGMA,OMEGA,IXMAX,NHARM)
              wax=wax+dwax
          END IF
C
      IF (BCOUNT.LE.NUMBE) THEN
          IF (WBEAM(BCOUNT).LE.SIGMA) THEN
              CALL BEWRI(WINDWX,SIGMA,OMEGA,IXMAX,NHARM)
              BCOUNT = BCOUNT + 1
          END IF
      END IF
C
      IF (SIGMA.GE.STOP) THEN
          GO TO 999
      END IF
C
C   End Richtmyer loop.
C
        GO TO 105
C
C***END KZKCYL
C
999   RETURN
      END
      SUBROUTINE TRIDIA(A,IXMAX,W,F,L,U)
C***BEGIN PROLOGUE TRIDIA
C***AUTHOR Jarle Berntsen, University of Bergen,
C          EDB-senteret, Herman Fossgt 6, 5007 Bergen, NORWAY.
C***DATE WRITTEN 880810
C***PURPOSE
C       BLOTRI SOLVES A BLOCK-THREE DIAGONAL SYSTEM OF EQUATIONS
C       BY BLOCK GAUSSIAN ELIMINATION.
C       40*IXMAX OPERATIONS ARE NEEDED.
C
C       THE BLOCKS ARE 2*2.
C       THE 2*2 SUB-MATRICES BELOW AND ABOVE THE MAIN DIAGONAL OF
C       A MUST HAVE ZEROS ON THEIR DIAGONALS.
C
C***DESCRIPTION
C
C       ON ENTRY
C               A        DOUBLE PRECISION(2,2,-1:1,IXMAX).
C                        DEFINES THE
C                        THREE-DIAGONAL MATRIX.
C                        THE FIRST TWO INDEXES DEFINES THE
C                        POSITION OF
C                        AN ELEMENT WITHIN A 2*2 SUB-MATRIX.
C                        THE THIRD INDEX DEFINES THE DISTANCE
C                        OF THE
C                        SUB-MATRIX FROM THE DIAGONAL.
C                        THE LAST INDEX DEFINES THE ROW NUMBER
C                        OF THE SUB-MATRIX.
C                        THIS INDEX CONVENTION IS ALSO USED FOR
C                        L AND U. UNCHANGED ON EXIT.
C
C               IXMAX    INTEGER. THE NUMBER OF DIAGONAL
C                        SUB-MATRICES.
C
C               F        DOUBLE PRECISION(2,IXMAX). DEFINES THE
C                        RIGHT-HAND SIDE. MODIFIED ON EXIT.
C
C               L        DOUBLE PRECISION(2,2,-1:0,IXMAX).
C                        WORKING SPACE
C                        USED FOR THE SUBMATRICES ON AND BELOW
C                        THE DIAGONAL
C                        OF THE L-MATRIX OF THE LU-DECOMPOSITION.
C
C               U        DOUBLE PRECISION(2,2,1:1,IXMAX). WORKING
C                        SPACE USED FOR THE SUBMATRICES ABOVE
C                        THE DIAGONAL OF
C                        THE U MATRIX OF THE LU-DECOMP.
C                        THE DIAGONAL SUBMATRICES OF U ARE
C                        UNIT 2*2.
C
C       ON RETURN
C
C               W        DOUBLE PRECISION(2,IXMAX).
C                        CONTAINS THE SOLUTION.
C
C***ROUTINES CALLED NONE
C***END PROLOGUE TRIDIA
C
```

```
C  Global variables.
C
      INTEGER IXMAX
      DOUBLE PRECISION A(2,2, - 1:1,IXMAX)
      DOUBLE PRECISION W(2,IXMAX),F(2,IXMAX),L(2,2, - 1:0,IXMAX)
      DOUBLE PRECISION U(2,2,1:1,IXMAX)
C
C  Local variables.
C
      INTEGER I,J
      DOUBLE PRECISION F1,F2,L21,U22
C
C       LU-DECOMPOSE.
C
C       FIRST STEP. (FIRST ROW)
C
C       L(0,1)=A(0,1)
C
C***FIRST EXECUTABLE STATEMENT TRIDIA
C
      DO 1 I = 1,2
          DO 1 J = 1,2
1     L(I,J,0,1) = A(I,J,0,1)
C
C       THE 2*2 SYSTEMS ARE SOLVED AS FOLLOWS:(AX=F)
C       L21=A21/A11
C       U22=A22-L21*A12
C       F2=F2-L21*F1
C       X2=F2/U22
C       X1=(F1-A12*X2)/A11
C
C       L(0,1)*U(1,1)=A(1,1)
C
      L21 = L(2,1,0,1)/L(1,1,0,1)
      U22 = L(2,2,0,1) - L21*L(1,2,0,1)
C       F2=A(2,1,1,1)-L21*A(1,1,1,1)(WE EXPLOIT ALL ZEROS)
      F2 = A(2,1,1,1)
      U(2,1,1,1) = F2/U22
C       U(1,1,1,1)=(A(1,1,1,1)-L(1,2,0,1)*U(2,1,1,1))/L(1,1,0,1)
      U(1,1,1,1) = - L(1,2,0,1)*U(2,1,1,1)/L(1,1,0,1)
C
      F1 = A(1,2,1,1)
      F2 = - L21*F1
      U(2,2,1,1) = F2/U22
      U(1,2,1,1) = (F1-L(1,2,0,1)*U(2,2,1,1))/L(1,1,0,1)
C
C       SECOND AND GENERAL STEP.
C
      DO 10 I = 2,IXMAX - 1
C
C       L(-1,I)=A(-1,I)
C
C         L(1,1,-1,I)=0
C         L(2,2,-1,I)=0
          L(1,2, - 1,I) = A(1,2, - 1,I)
          L(2,1, - 1,I) = A(2,1, - 1,I)
C
C       L(0,I)=A(0,I)-L(-1,I)*U(1,I-1)
C
          L(1,1,0,I) = A(1,1,0,I) - L(1,2, - 1,I)*U(2,1,1,I-1)
          L(1,2,0,I) = A(1,2,0,I) - L(1,2, - 1,I)*U(2,2,1,I-1)
          L(2,1,0,I) = A(2,1,0,I) - L(2,1, - 1,I)*U(1,1,1,I-1)
          L(2,2,0,I) = A(2,2,0,I) - L(2,1, - 1,I)*U(1,2,1,I-1)
C
C       L(0,I)*U(1,I)=A(1,I)
C
          L21 = L(2,1,0,I)/L(1,1,0,I)
          U22 = L(2,2,0,I) - L21*L(1,2,0,I)
          F2 = A(2,1,1,I)
          U(2,1,1,I) = F2/U22
          U(1,1,1,I) = - L(1,2,0,I)*U(2,1,1,I)/L(1,1,0,I)
          F1 = A(1,2,1,I)
          F2 = - L21*F1
          U(2,2,1,I) = F2/U22
          U(1,2,1,I) = (F1-L(1,2,0,I)*U(2,2,1,I))/L(1,1,0,I)
10    CONTINUE
C
C       END MAIN LOOP IN LU-DECOMP.
C
C       THIRD AND LAST STEP.(LAST ROW)
C
C       L(-1,IXMAX)=A(-1,IXMAX)
C
C       L(1,1,-1,IXMAX)=0
C       L(2,2,-1,IXMAX)=0
      L(1,2, - 1,IXMAX) = A(1,2, - 1,IXMAX)
      L(2,1, - 1,IXMAX) = A(2,1, - 1,IXMAX)
C
```

```
C       L(0,IXMAX)=A(0,IXMAX)-L(-1,IXMAX)*U(1,IXMAX-1)
C
      L(1,1,0,IXMAX) = A(1,1,0,IXMAX) -
     +                 L(1,2, - 1,IXMAX)*U(2,1,1,IXMAX-1)
      L(1,2,0,IXMAX) = A(1,2,0,IXMAX) -
     +                 L(1,2, - 1,IXMAX)*U(2,2,1,IXMAX-1)
      L(2,1,0,IXMAX) = A(2,1,0,IXMAX) -
     +                 L(2,1, - 1,IXMAX)*U(1,1,1,IXMAX-1)
      L(2,2,0,IXMAX) = A(2,2,0,IXMAX) -
     +                 L(2,1, - 1,IXMAX)*U(1,2,1,IXMAX-1)
C
C       END LU FACTORISATION.
C
C       FORWARD SUBSTITUTION.
C
C       FIRST STEP.
C
      L21 = L(2,1,0,1)/L(1,1,0,1)
      U22 = L(2,2,0,1) - L21*L(1,2,0,1)
      F(2,1) = F(2,1) - L21*F(1,1)
      F(2,1) = F(2,1)/U22
      F(1,1) = (F(1,1)-L(1,2,0,1)*F(2,1))/L(1,1,0,1)
C
C       SECOND AND GENERAL STEP.
C
      DO 20 I = 2,IXMAX
          L21 = L(2,1,0,I)/L(1,1,0,I)
          U22 = L(2,2,0,I) - L21*L(1,2,0,I)
          F1 = F(1,I) - L(1,2, - 1,I)*F(2,I-1)
          F2 = F(2,I) - L(2,1, - 1,I)*F(1,I-1)
          F2 = F2 - L21*F1
          F(2,I) = F2/U22
          F(1,I) = (F1-L(1,2,0,I)*F(2,I))/L(1,1,0,I)
20    CONTINUE
C
C       END FORWARD SUBSTITUTION.
C
C       BACK SUBSTITUTION.
C
C       FIRST STEP.
C
      W(1,IXMAX) = F(1,IXMAX)
      W(2,IXMAX) = F(2,IXMAX)
C
C       SECOND AND GENERAL STEP.
C
      DO 30 I = IXMAX - 1,1, - 1
          W(1,I) = F(1,I) - U(1,1,1,I)*W(1,I+1)
     +             - U(1,2,1,I)*W(2,I+1)
          W(2,I) = F(2,I) - U(2,1,1,I)*W(1,I+1)
     +             - U(2,2,1,I)*W(2,I+1)
30    CONTINUE
C
C       END BACK SUBSTITUTION.
C
C***END TRIDIA
C
      RETURN
      END

include 'itplbv.f'
```

mbe_tefb.f

Program for løsning av MBE og TEFB.

```
      program mbetbe
      implicit logical (a-z)
C
C
C     Written by Bjornar Ystad in 1994.
C
C*** purpose
C
C      This program simulates the sound field that is generated by
C      a source that is shaped as a spherical cap, and mounted in
C      a baffle.  This is done by solving the Mixed beam equation
C      and the Transformed beam equations numerically.
C      The curved boundary condition is treated by the Mixed beam
C      eqution.  Then certain transformations are performed that
C      transforms this solution into boundary conditions for the
C      prefocal version of the Transformed beam equation.  These two
C      equations give the sound field in the prefocal area.
C      In the focal plane the solution of the prefocal version of
C      the Transformed beam equation is transformed into boundary
C      conditions of the postfocal version of the Transformed beam
C      equation, and the sound field of the postfocal area is
C      obtained from the solution of the postfocal version of the
C      Transformed beam equation.
C
C*** on entry
C
C
C              actstart          double precision, the position of
C                                the center of the source, measured
C                                in units of the focal length.
C
C              ostart            double precision, the starting
C                                point for the solution of the
C                                Mixed beam equation.
C
C              actstop           double precision, the point where
C                                one starts storing data for the
C                                interpolation when solving the
C                                Mixed beam equation.
C
C              dsigma            double precision, the width in
C                                which data are stored for the
C                                interpolation.
C
C              fpoint            double precision, is the position
C                                of the focal plane.
C
C              fend              double precision, is the position
C                                of the end point of the procedure
C                                in which the postfocal version of
C                                the Wide angle equation is solved.
C
C*** routines called
C
C              widecyl, ph_shift, spline_2d_n, amp_trans_n
C***
C
C       variables that are used (mainly) in widesolve
C
      integer i
      double precision axl,incr,ostart,fpoint,fend,radius,
     +                actstart,actstop,start,stop,k,
     +                gain,r0,rd,freq,alpha2,sspeed,
     +                beta,mach
      character*4 ax

      include 'incl.f'
C
C     the first statements
C

      pi = 4.0*atan(1.d0)
C
C     SETTING THE PHYSICAL PARAMETERS
C

      OPEN(UNIT=9,FILE='innfil',STATUS='old')
3     format(a4,7x,e16.10,7x,e16.10,10x,e16.10)
      READ(9,3) ax,mach,flength,beta
      close(9)

C     Focal length [m]
c      flength = 1000.d-3
C     Source radius [m]
      radius = 24.3d-3
C     Frequency [Hz]
      freq = 1.1d6
C     Mach number
c      mach = 1.d-5
C     Sound speed [m/s]
      sspeed = 1490.0
C     Absorption coefficient [Np s^2/m]
      alpha2 = 2.5e-15
C     Nonlinearity parameter
c      beta = 3.5
C     Parameters that may be used for plots
      psource = 4.84
C
C     Wave number
      k = 2*pi*freq/sspeed
C     Rayleigh distance
      r0 = k*radius**2/2.d0
C     Plane wave shock distance
      rd = 1/(mach*beta*k)
C     Wave number times focal length
      kd = k*flength
C     Aperture angle
```

```
      soangle = radius/flength
      oangle = asin(soangle)
C     Absorption parameter
      pa = alpha2*freq**2*flength
C     Diffraction parameter
      pd = 1.d0/dble(2*kd)
C     Nonlinearity parameter
      pn = r0/rd
C
C     SETTING GEOMETRICAL PARAMETERS
C
C     Step length for the Richtmyer procedure
      rstep = 0.001
C     Intersection of the plane sigma = -1 and the cone satisfying
C     u = uwidth
C      ksimax = 4.0
      ksimax = 8.0
C     Distance between the focal point and the origin of the
C     TBE coordinates
C      tbedelta = 5.0
      tbedelta = 1.0
C     Position of the center of the source
      actstart = -1.0
C     Starting point for the MBE
C      ostart = -1.0*cos( oangle)
      ostart = -1.0
C     Starting point for storing interpolation data
      actstop = -0.7
C      actstop = -0.5
C     Position of the focal point
      fpoint = 0.0
C     End point of the calculations
      fend = 1/flength
C      fend = 0.5
C
C     Length of the interpolation region
      dsigma = (int(-actstop*(1-cos(oangle))/rstep) +1)*rstep
      uwidth = ksimax/(1+tbedelta)
C
C     INITIALIZING CONTROL VARIABLES
C
      nstep =0
      acount = 1
      bcount = 1
      max_energy =0.0
C
C     the points where axis data are to be stored
C
      axl = fend - actstart
      incr = axl/nant
      axl = actstart
      do 10 i=1, nant
 waxis(i) = axl
 axl = axl + incr
10    continue
C
C     the points where off-axis data are to be stored
C
      axl = fend - actstart
      incr = axl/bnant
      axl = actstart
      do 20 i=1,bnant
        axl = axl + incr
        bwaxis(i) = axl
20    continue
      do 30 i=1,bnant
        no_fc(i) = 200
30    continue
C
C     OPENING FILES
C
C     Axis plot
      open(unit=filn,file=ax)
C     Logging of the use of the Shapiro filter
C      logg_f_n = filn + 1
C      open(unit=logg_f_n,file='logg_f')
C     Off-axis plots
      transv_f_n = filn + 2
      open(unit=transv_f_n,file='tr_b')
C     List of parameters
      desc_f_n = filn + 3
      open(unit=desc_f_n,file='desc')
C
C     write parameters to file and standard output
C
      write(*,*) 'flength =',flength,' d =',radius
      write(*,*) 'f =',freq,' mach =',mach
      write(*,*) 'k =',k,' r0 =',r0,' rd =',rd
```

```
      write(*,*) 'pn =',pn,' pd =',pd,' pa =',pa
      write(desc_f_n,*) 'flength =',flength,' d =',radius
      write(desc_f_n,*) 'f =',freq,' mach =',mach
      write(desc_f_n,*) 'k =',k,' r0 =',r0,' rd =',rd
      write(desc_f_n,*) 'pn =',pn,' pd =',pd,' pa =',pa
C
C *** CALCULATIONS
C
C     the boundary condition of the Mixed beam equation
C
      call bound()
C
C     solving the Mixed beam equation
C
      mbflag = 1
      delta = 0.0
      start = ostart
      stop = actstop + dsigma
      call widecyl(start,stop,actstop)
C
C     performing the interpolation in the zeta1-phimbe plane
C
      delta = tbedelta
      call spline_2d_n()
C
C     performing the phase shift and the amplitude shift to
C     find boundary conditions for the prefocal version of the
C     Transformed beam equation
C
      call ph_shift_n()
      call amp_trans_n()
C
C     source conditions for using only the wide angle equation
C
C      call uni_bound(omega,ixmax,nharm,delta,uwidth)
C
C      gain = 0.5*kd*soangle**2
C      call gauss_bound(omega,ixmax,nharm,delta,uwidth,
C    +                 gain)
C
C
C     solving the prefocal version of the Transformed beam equation
C
      mbflag = 2
      pd = pd/(soangle**2)
      oangle = 0.0

      start = firstsigma
      stop = fpoint
      delta = tbedelta
      call widecyl(start,stop,actstop)
C
C     A second transformation is accomplished to find the
C     boundary conditions for the postfocal version of the
C     Transformed beam equation
C
      call fpl_transf()
C
C     solving the postfocal version of the
C     Transformed beam equation
C
      start = fpoint
      stop = fend
      delta = -tbedelta
      call widecyl(start,stop,actstop)
      end


      subroutine widecyl(start,stop,actstop)
      implicit logical (a-z)
C
C*** purpose
C
C      This routine solves the equations for the Fourier coefficients
C      of the Mixed beam equation and the Transformed beam equation.
C      It uses the numerical techniques called the fully implicit
C      method and Richtmyer's method.
C
C*** on entry
C
C            start       double precision, is the first value of the
C                        evolution variable
C
C            stop        double precision is the final value of the
C                        evolulution variable.
C
C            actstop     double precision is the position where one
C                        starts to store data for the interpolation
```

```
C                       when the Mixed beam equation is solved.
C** routines called
C
C           calccos, coeff, nonlin, output
C
C***
C
C     global variables
C
      double precision start,stop,actstop
C
      include 'incl.f'
C     local variables
C

      integer l, i, s,  is, i1, i2, filter

      double precision step, sig, oldsig, sigma, r, diag, offdia,
     +                istep, xstep, sigc, sigk, fakt1, fakt2,
     +                hres, hdeno, hsigma, lower, coangd

      parameter(      s = 20)
C
C     the first statements
C
      sigma = start

      if(start .eq. stop) then
         go to 999
      endif
C
      if(mbflag .eq. 1) then
        coangd = -start*cos(oangle)
        lower = atan(ksimax*(actstop + dsigma - tbedelta)/
     +             ((1 + tbedelta)*(actstop + dsigma)))
C        lower = atan(ksimax*(coangd + delta - stop)/
C    +              ((coangd + delta)*(coangd - stop)))
        xstep = (lower + pi/2.d0)/dble(2*(ixmax+1))
        WRITE(*,*) 'LOWER: ', lower/(ixmax+1)
        WRITE(*,*) 'xstep: ', xstep
        WRITE(*,*) 'PI/2: ', pi/2.d0/(ixmax+1)
C        xstep = pi/dble(2*(ixmax+1))
      else
        xstep = uwidth/(ixmax+1)
      endif

      call shap_trans()
C
C     calculate a factor of the diffraction/absorption coefficients
C     that depend on the variable phi (x)
C
      call calccos()
C
C     solve the equation in impstp step by using the implicit method
C
      do 100 is = 1,impstp
C
C     different step values are chosen for the various equations
C
         if(mbflag .eq. 1) then        ! MBE
            step = 0.005
C            step = 0.02
         elseif(delta .gt. 0.0) then   ! TEFB 1
            step = 0.005
C            step = 0.0005
         else                          ! TEFB 2
c            step = 0.000
             step = 0.005
         endif

         istep = step
         sig = sigma + step
         oldsig = sigma

 sigma = min(sig,stop)

         step = sigma - oldsig
         sigc = sigma
         sigk = sigma
C
C        calculate the coefficients of the absorption/diffraction
C        terms
C
         call coeff(sigc,sigk)
C
C        calculate the coefficients of the nonlinear terms
C
         call nonlin(sigma)
```

```
 r = step/xstep**2
C
C        for each harmonic component, find an approximate solution
C        to its equation by the implicit method
C
 do 50 l=1,nharm
C
C           make the first block - row in the matrix
C
   diag = 1+step*cnew(0,l)
   offdia = 2*r*khalf(0,l)

   a(1,1,0,0) = diag
   a(1,2,0,0) = 2*offdia
   a(2,1,0,0) = -a(1,2,0,0)
   a(2,2,0,0) = diag

           a(1,2,1,0) = -2*offdia
   a(2,1,1,0) = -a(1,2,1,0)
C
C          make the rest of the block rows except the last one
C
   do 10 i=1,ixmax - 1
      diag = 1 + step*cnew(i,l)
      offdia = 2*r*khalf(i,l)

      a(1,2,-1,i) = -offdia*(1-1/dble(2*i))/2
      a(2,1,-1,i) = -a(1,2,-1,i)


      a(1,1,0,i) = diag
      a(1,2,0,i) = offdia
      a(2,1,0,i) = -offdia
      a(2,2,0,i) = diag


      a(1,2,1,i) = -offdia*(1+1/dble(2*i))/2
      a(2,1,1,i) = -a(1,2,1,i)
10         continue
C
C          make the last block row
C
           diag = 1+step*cnew(ixmax,l)
           offdia = 2*r*khalf(ixmax,l)

           a(1,2,-1,ixmax) = -offdia*(1-1/dble(2*ixmax))/2
           a(2,1,-1,ixmax) = -a(1,2,-1,ixmax)


           a(1,1,0,ixmax) = diag
           a(1,2,0,ixmax) = offdia
           a(2,1,0,ixmax) = -offdia
           a(2,2,0,ixmax) = diag
C
C          make the right hand side of the system of equations
C
           do 20 i=0,ixmax
              f(1,i) = omega(i,l,1) + step*nonli(i,l,1)
     f(2,i) = omega(i,l,2) + step*nonli(i,l,2)
20         continue
C
C          solve the linear system of equations
C
           i1 = 1
           i2 = i1 + 8*ixmax + 8
           call tridia(a,ixmax+1,w,f,work(i1),work(i2))
C
C          substitute the solution into omega
C
           do 30 i=0,ixmax
             omega(i,l,1) = w(1,i)
             omega(i,l,2) = w(2,i)
30         continue
50      continue
C
C       perform the damping of the solution on the border
C       and the smoothing of the Shapiro filter
C
        call dampsol(omega,ixmax,nharm)
        call shap_smo()
C
C       call output in case there is data to be stored
C
        call output(sigma,actstop)

        if (sigma .ge. stop) then
           go to 999
        endif
```

```
100   continue
C
C     go on solving the equation by Richtmyer's method
C
105   continue
         hsigma = 1.0+sigma
C
C        choose step length
C        (in the interpolation region step should be equal to rstep,
C        as rstep is used in spline_2d_n and for calculating dsigma)
C
         if(hsigma .ge. 0.9 .and. hsigma .le. 1.4) then
            step = rstep/10.d0
         else
            step = rstep
         endif
C         if(hsigma .ge. 1.5) then
C            pa = 0.0
C         endif
C
         sig = sigma + step
         oldsig = sigma
         sigma = min(sig,stop)
         step = sigma - oldsig
         sigc = sigma
         sigk = oldsig + step/2
C
C        for each step calculate the coefficients of the absorption,
C        diffraction and nonlinear terms
C
         call coeff(sigc,sigk)
         call nonlin(sigma)

         r = step/xstep**2
C
C        find an approximate solution to each harmonic component
C        by using Richtmyer's method
C
         do 150 l=1,nharm
C
C        make the matrix for the system of equations
C
C           make the first block - row
C
            diag = 1+step*cnew(0,l)/2
            offdia = r*khalf(0,l)

            a(1,1,0,0) = diag
            a(1,2,0,0) = 2*offdia
            a(2,1,0,0) = -a(1,2,0,0)
            a(2,2,0,0) = diag


            a(1,2,1,0) = -2*offdia
            a(2,1,1,0) = -a(1,2,1,0)


C
C           make the rest of the block rows except the last one
C

            do 110 i=1,ixmax - 1
               diag = 1 + step*cnew(i,l)/2
               offdia = r*khalf(i,l)

               a(1,2,-1,i) = -offdia*(1-1/dble(2*i))/2
               a(2,1,-1,i) = -a(1,2,-1,i)


               a(1,1,0,i) = diag
               a(1,2,0,i) = offdia
               a(2,1,0,i) = -offdia
               a(2,2,0,i) = diag


               a(1,2,1,i) = -offdia*(1+1/dble(2*i))/2
               a(2,1,1,i) = -a(1,2,1,i)
110         continue
C
C           make the last block row
C
            diag = 1+step*cnew(ixmax,l)/2
            offdia = r*khalf(ixmax,l)

            a(1,2,-1,ixmax) = -offdia*(1-1/dble(2*ixmax))/2
            a(2,1,-1,ixmax) = -a(1,2,-1,ixmax)


            a(1,1,0,ixmax) = diag
            a(1,2,0,ixmax) = offdia
            a(2,1,0,ixmax) = -offdia
            a(2,2,0,ixmax) = diag
C
C        make the right hand side of the system of equations
C
C        the first component of the right hand side


         diag = 1 - step*cold(0,l)/2
         offdia = r*khalf(0,l)

         f(1,0) = diag*omega(0,l,1) +
     +            2*offdia*(omega(1,l,2) - omega(0,l,2)) +
     +            step*nonli(0,l,1)

         f(2,0) = diag*omega(0,l,2) -
     +            2*offdia*(omega(1,l,1) - omega(0,l,1)) +
     +            step*nonli(0,l,2)
C
C        the components in between
C
         do 120 i=1,ixmax-1

            diag = 1 - step*cold(i,l)/2
            offdia = r*khalf(i,l)

            fakt1 = 1 + 1/dble(2*i)
            fakt2 = 1 - 1/dble(2*i)

            f(1,i) = diag*omega(i,l,1) +
     +            offdia*(fakt1*omega(i+1,l,2)-2*omega(i,l,2)+
     +            fakt2*omega(i-1,l,2))/2 +step*nonli(i,l,1)

            f(2,i) = diag*omega(i,l,2) -
     +            offdia*(fakt1*omega(i+1,l,1)-2*omega(i,l,1)+
     +            fakt2*omega(i-1,l,1))/2 +step*nonli(i,l,2)
120      continue
C
C        the last component of the right hand side
C
         diag = 1 - step*cold(ixmax,l)/2
         offdia = r*khalf(ixmax,l)
 fakt2 = 1 - 1/dble(2*ixmax)
         f(1,ixmax) = diag*omega(ixmax,l,1) +
     +             offdia*(fakt2*omega(ixmax-1,l,2) -
     +             2*omega(ixmax,l,2))/2+step*nonli(ixmax,l,1)

         f(2,ixmax) = diag*omega(ixmax,l,2) -
     +             offdia*(fakt2*omega(ixmax-1,l,1) -
     +             2*omega(ixmax,l,1))/2+step*nonli(ixmax,l,2)
C
C        solve the system of equations
C
         i1 = 1
         i2 = i1 + 8*ixmax + 8
         call tridia(a,ixmax+1,w,f,work(i1),work(i2))
C
C        substitute the solution into omega
C
         do 130 i=0,ixmax
            omega(i,l,1) = w(1,i)
            omega(i,l,2) = w(2,i)
130      continue
C
C        store the value of the absorption coefficient
C
         do 140 i=0,ixmax
            cold(i,l) = cnew(i,l)
140      continue
150   continue
C
C     perform the damping of the solution
C     and the smoothing of the Shapiro filter
C
      call dampsol(omega,ixmax,nharm)
      call test_fc_siz(filter)
      if(filter .eq. 1) then
         call shap_smo()
         hres = 1.0
      else
         hres = 0.0
      endif
      hsigma = 1.0 + sigma

C
C     register whether the filter has been used or not
```

```
C
C      write(logg_f_n,*) hsigma, hres
C
C     call output in case there is any data to be stored
C
      call output(sigma,actstop)

      if (sigma .ge. stop) then
         go to 999
      endif
      go to 105
999   continue
      return
      end


      subroutine test_fc_siz(filter)
      implicit logical(a-z)
C
C     This routine estimates the energy of the 10 highest harmonics
C     and compares it to the energy of the total solution.
C     This is used to decide whether the Shapiro filter is to
C     be used of not.
C
C     global variables
C
      integer filter
      include 'incl.f'
C
C     local variables
C
      integer i,j,k
      double precision up_energy,tot_energy,cmp

      tot_energy=0.0
      do 30 k=1,2
 do 20 j=1,nharm
    do 10 i=1,ixmax
       tot_energy=tot_energy+omega(i,j,k)**2
10          continue
20       continue
30    continue
      tot_energy=sqrt(tot_energy)
      if(tot_energy .gt. max_energy) then
         max_energy = tot_energy
      endif

      up_energy=0.0
      do 60 k=1,2
 do 50 j=nharm,nharm-10,-1
    do 40 i=1,ixmax
       up_energy=up_energy+omega(i,j,k)**2
40          continue
50       continue
60    continue
      up_energy=sqrt(up_energy)

      cmp=0.001*max_energy

      if(up_energy .ge. cmp) then
 filter = 1
      else
 filter = 0
      endif
      return
      end


      subroutine shap_trans()
      implicit logical(a-z)
C
C     This routine calculates the amplitude transfer function of
C     the Shapiro filter
C
C     global variables
C
      include 'incl.f'
C
C     local variables
C
      integer i,j,power

      double precision sarg,help,arg

      power = 2*ord_shap + 2
      shap(nharm) = 0.0

      do 20 j=1,nharm-1
         arg = (dble(j)/dble(nharm))*(pi/2.0)
         sarg = sin(arg)
         help = sarg
         do 10 i=1,power-1
            help = help*sarg
10       continue
         shap(j)=1.0-help
20    continue
      return
      end


      subroutine shap_smo()
      implicit logical (a-z)
C
C     This routine performs the smoothing of the Shapiro filter
C
C     global variables
C
      include 'incl.f'
C
C     local variables
C
      integer i,j,k

      do 30 k=1,2
         do 20 j=1,nharm
            do 10 i=0,ixmax
               omega(i,j,k) = omega(i,j,k)*shap(j)
10          continue
20       continue
30    continue
      return
      end


      subroutine calccos()
      implicit logical (a-z)
C
C*** purpose
C
C      The routine calculates the part of the absorption/diffraction
C      coefficients that depends upon the transverse variable
C***
C
      include 'incl.f'
C     local variables
C
      double precision angstep, coang, angle
      integer  border, i
C
C     first statements
C
      if (mbflag .eq. 1) then
C
C     the Mixed beam equation is solved
C
         angstep = pi/(2*(ixmax+1))
         border = nint(oangle/angstep)
         angle = 0.0
C
C        the values of the factors in the inner area
C        (if there is any)
C
         if(border .gt. 0) then
            coang = cos(oangle)
            do 10 i=0, border
               cc(i) = coang
               cinvc(i) = 1/coang
10          continue
            angle = (border+1)*angstep
         else
    cc(0) = 1.0
    cinvc(0) = 1.0
         endif
C
C        the values of the factors in the outer area
C
         if(border .lt. ixmax) then
            do 20 i=border+1,ixmax
               cc(i) = cos(angle)
               cinvc(i) = 1/cc(i)
               angle = angle + angstep
20          continue
         endif
      else
C
C     the Transformed beam equation is solved
```

```
C
         do 30 i=0,ixmax
            cc(i) = 1.0
            cinvc(i) = 1.0
30       continue
      endif
      return
      end


      subroutine coeff(sigc,sigk)
      implicit logical (a-z)
C
C*** purpose
C
C      The routine calculates coefficients for the diffraction/
C      absorption terms when the equations of this program are
C      solved by the implicit method and by Richtmyer's method.
C
C*** on entry
C
C           sigc        double precision, is the value of the
C                       evolution variable that is used in the
C                       calculation of the absorption coefficient.
C                       It corresponds to step "j+1" for both the
C                       fully implicit and
C                       the Richtmyer method.
C
C           sigk        double precision, is the value of the
C                       evolution variable that is used in the
C                       calculation of the diffraction coefficient.
C                       It corresponds to step "j+1" for the fully
C                       implicit and "j+1/2" for the Richtmyer
C                       method.
C
C
C*** routines called none
C
C     global variables
C
      double precision sigc,sigk
      include 'incl.f'
C     local variables
C
      integer i,j
C
      double precision hsigk
C     first statements
C
      hsigk = sigk-delta
C
      do 20 j = 1,nharm
 do 10 i=0,ixmax
    cnew(i,j) = pa*cinvc(i)*j**2
    khalf(i,j) = -pd*cc(i)/(j*hsigk**2)
10       continue
20    continue
      return
      end


      subroutine axwri(actsigma)
C
C*** purpose
C
C     This routine writes the pressure on axis in the position
C     corresponding to sigma to the file with ref. no. filn
C
C
C*** on entry
C
C            actsigma   double precision, is the evolution variable.
C
C*** routines called none
C
C
      implicit logical (a-z)
C
C     global variables
C
      double precision actsigma
      include 'incl.f'
C
C     local variables
C
      integer i,j
      double precision abrad,sigma,hsig,hres,htab(2*max),
     +                 hstep,deno
C
C     statements
C
      sigma = actsigma
      abrad = abs(sigma)+abs(delta)
      hstep = uwidth/(ixmax+1)
      deno = actsigma - delta

      hsig = sigma + 1
      do 10 j=1,max
        i=j
        htab(j) = omega(0,i,1)/deno
        htab(j+max) = omega(0,i,2)/deno
10    continue
      write(filn,1) hsig,(htab(j),j=1,2*max)
1     format(e16.10, 60(1x,e16.10))
      return
      end


      subroutine fcbewri(sigma)
      implicit logical (a-z)
C
C*** purpose
C
C     This routine writes the Fourier coefficients in
C     off-axis points to the file with ref. no. transv_f_n.
C
C*** on entry
C
C     sigma    double precision, the evolution variable.
C
C*** routines called none
C
C
C     global variables
C
      double precision sigma
      include 'incl.f'
C
C     local variables
C
      integer i,j,hind,indincr,hantsp
      double precision dksi,ksim,expincr,hdeno,abdeno,ustep,u,ex
      double precision ha(max),hb(max)
C
C     statements
C
      u = 0.0
      hantsp = no_fc(bcount)
      hdeno = sigma - delta
      abdeno = abs(hdeno)
      ksim = abdeno*uwidth
      dksi = ksim/dble(ixmax + 1)
      indincr = nint(expincr/dksi)
      if(indincr .eq. 0) then
        indincr = 1
      endif
      ustep = uwidth/(hantsp + 1)

      write(*,*) '-+------      ---      ---      ------+-'
      write(*,*) '   FCBEWRI, sigma =',sigma,','
      write(*,*) '   bwaxis(',bcount,') =',bwaxis(bcount),','
      write(*,*) '   indincr =',indincr,' uwidth =',uwidth,
     +          ' ksim =',ksim
      write(desc_f_n,*) '-+------      ---      ---      ------+-'
      write(desc_f_n,*) '   FCBEWRI, sigma =',sigma,','
      write(desc_f_n,*) '   bwaxis(',bcount,') =',bwaxis(bcount),','
      write(desc_f_n,*) '   indincr =',indincr,' uwidth =',uwidth,
     +                ' ksim =',ksim
      if(bnant .eq. bcount) then
        write(desc_f_n,*) '-+------      ---      ---      ------+-'
        write(*,*) '-+------      ---      ---      ------+-'
      endif

      do 20 i=0,hantsp-1
        ex = abdeno*u
        hind = indincr*i
        do 10 j=1,max
          ha(j) = omega(hind,j,1)/hdeno
          hb(j) = omega(hind,j,2)/hdeno
10      continue
        write(transv_f_n,2) ex,(ha(j),j=1,max),
     +                      (hb(j),j=1,max)
        u = u + ustep
20    continue
2     format(e16.10,60(1x,e16.10))
      return
```

```
      end


      subroutine nonlin(sigma)
      implicit logical (a-z)
C
C*** purpose
C
C       This routine calculates the nonlinear contributions to the
C       equations for the fourier coefficients.
C
C*** on entry
C
C           sigma       double precision, is the position in which
C                       the equation is currently being solved.
C
C
C
C*** routines called none
C
C
CC
CC      global variables
CC
      double precision sigma
      include 'incl.f'
C
C     local variables
C
      integer i,j,n,hmax,hant
      external ddot
      double precision fact, s, hu(nharm), hv(nharm),ddot,
     +              hw(nharm),hx(nharm),hy(nharm),hz(nharm)

C
C     statements
C
      s = sigma - delta
      if (mbflag .eq. 1) then
         s = abs(s)
      endif

      hmax=ixmax/2

      do 100 n=1,nharm
 do 20 i=0,ixmax
    nonli(i,n,1) = 0.0
    nonli(i,n,2) = 0.0
20       continue
         hant = nharm - n



         do 50 i=0,hmax
            do 30 j=1,n-1
               nonli(i,n,1) = nonli(i,n,1) +
     +                        omega(i,j,2)*omega(i,n-j,1)
C               hu(j) = omega(i,j,2)
C               hv(j) = omega(i,n-j,1)
               nonli(i,n,2) = nonli(i,n,2) +
     +                        omega(i,j,2)*omega(i,n-j,2)-
     +                        omega(i,j,1)*omega(i,n-j,1)
C               hw(j) = omega(i,n-j,2)
C               hx(j) = omega(i,j,1)
30          continue
C            nonli(i,n,1) = ddot(n-1,hu,1,hv,1)
C            nonli(i,n,2) = ddot(n-1,hu,1,hw,1)-ddot(n-1,hx,1,hv,1)
            nonli(i,n,2) = 0.5*nonli(i,n,2)
C
            do 40 j=n+1,nharm
               nonli(i,n,1) = nonli(i,n,1) +
     +                        omega(i,j-n,1)*omega(i,j,2)-
     +                        omega(i,j-n,2)*omega(i,j,1)
C               hu(j-n) = omega(i,j-n,1)
C               hv(j-n) = omega(i,j,2)
C               hw(j-n) = omega(i,j-n,2)
C               hx(j-n) = omega(i,j,1)

               nonli(i,n,2) = nonli(i,n,2) -
     +                        omega(i,j-n,1)*omega(i,j,1) -
     +                        omega(i,j-n,2)*omega(i,j,2)
40          continue
C               nonli(i,n,1) = nonli(i,n,1) +
C     +                        ddot(hant,hu,1,hv,1)-
C     +                        ddot(hant,hw,1,hx,1)
C               nonli(i,n,2) = nonli(i,n,2) -
C     +                        ddot(hant,hu,1,hx,1)-
C     +                        ddot(hant,hw,1,hv,1)
50       continue

 fact = pn*n/s
         do 60 i=0,ixmax
    nonli(i,n,1) = fact*nonli(i,n,1)
    nonli(i,n,2) = fact*nonli(i,n,2)
60       continue

100   continue
      return
      end


      subroutine bound()
      implicit logical (a-z)
C
C*** purpose
C
C       This routine assigns the boundary condition of the Mixed beam
C       equation to the quantity omega.
C
C*** routines called none
C
C     global variables
C
      include 'incl.f'
C
C     local variables
C
      integer i,j,border
      double precision angstep
C
C     statements
C
      angstep = pi/(2*(ixmax+1))
      border = nint(oangle/angstep)
C
C     the harmonic variation on the active source
C
      if (border .gt. 0) then
 do 10 i=0,border
    omega(i,1,1) = 0.0
    omega(i,1,2) = 1.0
10       continue
      endif
C
C     the first harmonics on the baffle
C
      if (border .lt. ixmax) then
 do 20 i=border+1,ixmax
    omega(i,1,1) = 0.0
    omega(i,1,2) = 0.0
20    continue
      endif
C
C     the rest of the harmonics
C
      do 40 j=2,nharm
         do 30 i=0,ixmax
            omega(i,j,1) = 0.0
            omega(i,j,2) = 0.0
30       continue
40    continue
      return
      end


      subroutine output(sigma,actstop)
      implicit logical (a-z)

C
C*** purpose
C
C       The purpose of this routine is:
C       1) When the Mixed beam equation is solved it stores data for
C       the interpoation.
C       2) It administers the writing of axis plots.
C
C***
C
C           sigma       double precision, is the position of the
C                       current solution.
C
C           actstop     double precision, is the position where we
C                       will start storing solutions for the
C                       interpolation when the Mixed beam equation
C                       is solved.
C*** routines called
```

```
C
C      wr_polgrid, axwri
C
C***
C
C     global variables
C
      double precision sigma,actstop
      include 'incl.f'
C     local variable
C
      double precision actsigma
C
C     statements
C
C     When we are solving the Mixed beam equation, the solutions
C     from actstop to the final solution are stored in polgrid
C     to make the coming interpolation possible. firstsigma is
C     the sigma value of the first solution that is stored in
C     polgrid.
C

      if (mbflag .eq. 1) then
 if (sigma .ge. actstop) then
    nstep = nstep+1
          call wr_polgrid()
        endif
 if (nstep .eq. 1) then
    firstsigma = sigma
        endif
      endif
C
C     storing of data for axis plots
C
      if (mbflag .eq. 1) then
 actsigma = sigma/cos(oangle)
      else
 actsigma = sigma
      endif

      if (acount .le. nant) then
         if (waxis(acount) .le. actsigma) then
    if (mbflag .eq. 1) then
       if (actsigma .lt. actstop) then
                call axwri(actsigma)
                acount = acount+1
             endif
          else
             call axwri(actsigma)
             acount=acount+1
    endif
        endif
      endif

      if(bcount .le. bnant) then
        if(bwaxis(bcount) .le. actsigma) then
          if(mbflag .eq. 1) then
            if(actsigma .lt. actstop) then
              call fcbewri(actsigma)
              bcount = bcount + 1
            endif
          else
            call fcbewri(actsigma)
            bcount = bcount + 1
          endif
        endif
      endif
      return
      end


      subroutine wr_polgrid()
      implicit logical (a-z)
C
C*** purpose
C      The routine writes the solution omega to polgrid.
C
C*** routines called none
C
C
      include 'incl.f'
C     local variables
C
      integer i,j,k
C
C     statements
C
      if(nstep .gt. sigdim) then
         write(*,*) 'Array index out of bounds in wr_polgrid,'
         write(*,*) 'increase sigdim to match nstep: ',nstep
         write(*,*) '(making more room in polgrid)'
         stop
      endif
      do 30 k=1,2
 do 20 j = 1,nharm
    do 10 i = 0,ixmax
       polgrid(i,j,k,nstep) = omega(i,j,k)
10         continue
20      continue
30    continue
      return
      end


      subroutine ph_shift_n()
C
C**purpose
C
C  When there  is a change in the definition of the retarded time
C  a phase shift must be performed.  This routine performs the
C  phase shift in the transition between the Mixed beam equation
C  and prefocal version the Transformed beam equation.
C
C**routines called none
C
      implicit logical (a-z)

C     global variables
C
      include 'incl.f'
C     local variables
C
      integer i,k
      double precision ustep,u,oldangle,coldangle,sigma,
     +          phase,ckph,skph,newa,newb
C
C     first statements
C
      ustep = uwidth/(ixmax+1)
      u     = 0.0
      sigma=firstsigma

C
C     Perform the phase transformation
C

      do 20 i=0,ixmax
C
C        for each angle in the new coordinate system, find the
C        corresponding angle in the old one, and find the change
C        time definition
C
         oldangle = atan((1+abs(delta/sigma))*soangle*u)
         coldangle = cos(oldangle)
         phase = -kd*((1+sigma)+0.5*(soangle*u)**2*(sigma-delta)-
     +            sigma/coldangle)

         do 10 k=1,nharm
C
C        find the phaseshift for each harmonic
C
            ckph=cos(k*phase)
            skph=sin(k*phase)
            newa=omega(i,k,1)*ckph-omega(i,k,2)*skph
            newb=omega(i,k,2)*ckph+omega(i,k,1)*skph
            omega(i,k,1)=newa
            omega(i,k,2)=newb
10       continue
C
C           increment the transverse variable
C
         u = u+ustep
20    continue
      return
      end


      subroutine amp_trans_n()
C
C**purpose
C
C     The amplitude transformation that is performed by this routine,
C     is one of the transformations that must be performed to find
C     boundary conditions for the prefocal version of the
C     Transformed beam equation
C     from the solution of the Mixed beam equation.
```

```
C
C***routines called none
C
      implicit logical (a-z)
C
C     global variables
C
      include 'incl.f'
C     local variables
C
      integer i,j
      double precision oldangle,u,ustep,fact,c1,c2
C
C     statements
C
      ustep = uwidth/(ixmax+1)
      u = 0.0
      c1 = 1.0+abs(delta/firstsigma)*sin(oangle)
      c2 = -(1.0+abs(delta/firstsigma))
      do 20 j=1,nharm
         do 10 i=0,ixmax
            oldangle = atan(c1*u)
            fact = c2*cos(oldangle)
            omega(i,j,1)=fact*omega(i,j,1)
            omega(i,j,2)=fact*omega(i,j,2)
            u = u+ustep
10       continue
20    continue
      return
      end


      subroutine spline_2d_n()
C
C***purpose
C   The routine performs the interpolation in the phimbe-zeta1
C   plane to find boundary conditions for the prefocal version
C   of the Transformed beam equation from the solution of the Mixed
C   beam equation.  It uses the ACM routine itplbv.f
C   for the twodimensional spline interpolation.
C
C****
C
      implicit logical (a-z)
C
      include 'incl.f'
C
C    local variables
C
      integer i,j,k,l,border,t
      double precision angstep,angle,sigma,fact,oldangle,
     +                 u,ustep,sthet
C
C     statements
C
C
C     make the tables for the coordinates of the vertices in
C     the "zeta1-phimbe" plane
      write(*,*) 'spline_2d_n; nstep firstsigma pi rstep'
      write(*,*) nstep,firstsigma,pi,rstep

      angstep = pi/(2*(ixmax+1))
      angle = 0.0
      do 10 i=0,ixmax
 x(i+1)=angle
 angle = angle+angstep
10    continue
      x(ixmax+2)=pi/2.0

      sigma = firstsigma
      do 20 i = 1,nstep
 y(i) = sigma
 sigma = sigma + rstep
20    continue
C
C     find the points in the  "zeta1-phimbe" plane where we want to
C     know the value of omega
C
      border = nint(oangle/angstep)
      fact = 1 + delta/abs(firstsigma)
      u = 0.0
      ustep = uwidth/(ixmax+1)
      sthet = sin(oangle)
      do 30 i=0,ixmax+1
 oldangle = atan(fact*sthet*u)
 tx(i+1) = oldangle
 u = u+ustep
30    continue
```

```
      fact=cos(oangle)*firstsigma
      do 40 i=0,ixmax+1
 if( tx(i+1) .lt. oangle) then
    ty(i+1)= fact/cos(tx(i+1))
        else
    ty(i+1)=firstsigma
        endif
40    continue
C
C     do the interpolation for each Fourier coefficient
C
      do 100 j=1,nharm
 do 90 k=1,2

    do 70 i=0,ixmax
       do 60 l=1,nstep
          fi(i+1,l)=(polgrid(i,j,k,l))
60           continue
70          continue
            do 75 l=1,nstep
               fi(mx,l) = 0.0
75          continue

C            call itplbv(6,mx,nstep,x,y,fi,n,tx,ty,ff)
            call itplbv(6,mx,nstep,x,y,fi,ixmax+2,tx,ty,ff)

    do 80 i=0,ixmax
       omega(i,j,k) = dble(ff(i+1))
80          continue
90       continue
100   continue
      return
      end


      include 'itplbv.f'


      subroutine fpl_transf()
      implicit logical (a-z)
C
C**description
C
C   This routine performs the transformation that is necessary to
C   find the boundary conditions of the postfocal version of the
C   Transformed beam equation from the solution of the prefocal
C   version of the Transformed beam equation in the focal plane.
C
C***routines called none
C
C     global variables
C
C

      include 'incl.f'
C     local variables
C
      integer i,j,k
      double precision u,ustep,fact,phase,ckph,skph,newa,newb
C
C     first statements
C
      ustep = uwidth/(ixmax+1)
      u = 0.0
C
C     perform the phase - transformation
C
      fact = -kd*delta*soangle**2
C
      do 20 i=0,ixmax

         phase = fact*u**2

         do 10 k=1,nharm
C
C        find the phaseshift for each harmonic
C
            ckph=cos(k*phase)
            skph=sin(k*phase)
            newa=omega(i,k,1)*ckph-omega(i,k,2)*skph
            newb=omega(i,k,2)*ckph+omega(i,k,1)*skph
            omega(i,k,1)=newa
            omega(i,k,2)=newb
10       continue
C
C           increment the transverse variable
C
         u = u+ustep
```

```
20    continue
C
C     the (very simple) amplitude transformation
C
      do 50 k=1,2
         do 40 j=1,nharm
            do 30 i=0,ixmax
               omega(i,j,k)=-omega(i,j,k)
30          continue
40       continue
50    continue
      return
      end


      subroutine uni_bound(omega,ixmax,nharm,delta,uwidth)
      implicit logical (a-z)
C
C     global variables
C
      integer ixmax,nharm
      double precision omega(0:ixmax,nharm,2),delta,uwidth
C
C     local variables
C
      integer i,j,k,border
      double precision u,ustep,ulim,ksilim,gamma

      gamma=1.0+delta
      ksilim = 1.0
      ulim = ksilim/gamma
      ustep = uwidth/(ixmax+1)
      u=0.0

      border = nint(ulim/ustep)
C
C     values on the active source
C
      do 10 i=0,border
 omega(i,1,1)=0.0
 omega(i,1,2)=1.0*gamma
10    continue
C
C     values on the baffle
C
      do 20 i=border+1,ixmax
         omega(i,1,1)=0.0
 omega(i,1,2)=0.0
20    continue
C
C     the coefficients of the higher harmonics equal zero
C
      do 50 k=1,2
 do 40 j=2,nharm
    do 30 i=0,ixmax
       omega(i,j,k)=0.0
30          continue
40       continue
50    continue
      return
      end


      subroutine gauss_bound(omega,ixmax,nharm,delta,uwidth,
     +                       gain )
      implicit logical (a-z)
C
C     global variables
C
      integer ixmax,nharm
      double precision omega(0:ixmax,nharm,2),delta,uwidth,
     +                 gain
C
C     local variables
C
      integer i,j,k,border
      double precision u,ustep,ulim,ksilim,gamma,fact

      gamma=1.0+delta
      ksilim = 1.0
      ulim = ksilim/gamma
      fact = gain*gamma*delta
      ustep = uwidth/(ixmax+1)
      u=0.0
      border = nint(ulim/ustep)
C
C     values on the active source
C
```

```
      do 10 i=0,border
         omega(i,1,1)=-gamma*sin(fact*u**2)
         omega(i,1,2)=-gamma*cos(fact*u**2)
         u = u+ustep
10    continue
C
C     values on the baffle
C
      do 20 i=border+1,ixmax
         omega(i,1,1)=0.0
         omega(i,1,2)=0.0
20    continue
C
C     the coefficients of the higher harmonics equal zero
C
      do 50 k=1,2
         do 40 j=2,nharm
            do 30 i=0,ixmax
               omega(i,j,k)=0.0
30          continue
40       continue
50    continue
      return
      end


      subroutine dampsol(omega,ixmax,nharm)
      implicit logical (a-z)
C
C     global variables
C
      integer ixmax,nharm,hww
      double precision omega(0:ixmax,nharm,2)
C
C     local variables
C
      integer i,j,k,hind
      double precision dfact(0:999),harg,alpha(8000)


      hww =2*(ixmax/5)
      do 11 i=1,hww
         alpha(i) =(dble(hww-i+1)/dble(hww))**2
11    continue

C      do 10 i=1,400
C         alpha(i) =((400.0 -i+1.0)/400.0)**2
C10    continue
C
      do 15 i=0,hww-1
         dfact(i) = 1.0 - alpha(i+1)
15    continue
C
      do 40 k=1,2
 do 30 j=1,nharm
    do 20 i=0,hww-1
       hind = ixmax-i
               omega(hind,j,k) = omega(hind,j,k)*dfact(i)
20          continue
30       continue
40    continue
      return
      end


      SUBROUTINE TRIDIA(A,IXMAX,W,F,L,U)
C***BEGIN PROLOGUE TRIDIA
C***AUTHOR Jarle Berntsen, University of Bergen,
C          EDB-senteret, Herman Fossgt 6, 5007 Bergen, NORWAY.
C***DATE WRITTEN 880810
C***PURPOSE
C       BLOTRI SOLVES A BLOCK-THREE DIAGONAL SYSTEM OF EQUATIONS BY
C       BLOCK GAUSSIAN ELIMINATION.
C       40*IXMAX OPERATIONS ARE NEEDED.
C
C       THE BLOCKS ARE 2*2.
C       THE 2*2 SUB-MATRICES BELOW AND ABOVE THE MAIN DIAGONAL OF A MUST
C       HAVE ZEROS ON THEIR DIAGONALS.
C
C***DESCRIPTION
C
C       ON ENTRY
C              A        DOUBLE PRECISION(2,2,-1:1,IXMAX). DEFINES
C                       THE THREE-DIAGONAL MATRIX.
C                       THE FIRST TWO INDEXES DEFINES THE POSITION
C                       OF AN ELEMENT WITHIN A 2*2 SUB-MATRIX.
C                       THE THIRD INDEX DEFINES THE DISTANCE OF
C                       THE SUB-MATRIX FROM THE DIAGONAL.
```

```
C                        THE LAST INDEX DEFINES THE ROW NUMBER OF
C                        THE SUB-MATRIX.
C                        THIS INDEX CONVENTION IS ALSO USED FOR L
C                        AND U. UNCHANGED ON EXIT.
C
C               IXMAX    INTEGER. THE NUMBER OF DIAGONAL
C                        SUB-MATRICES.
C
C               F        DOUBLE PRECISION(2,IXMAX). DEFINES THE
C                        RIGHT-HAND SIDE. MODIFIED ON EXIT.
C
C               L        DOUBLE PRECISION(2,2,-1:0,IXMAX).
C                        WORKING SPACE
C                        USED FOR THE SUBMATRICES ON AND BELOW THE
C                        DIAGONAL
C                        OF THE L-MATRIX OF THE LU-DECOMPOSITION.
C
C               U        DOUBLE PRECISION(2,2,1:1,IXMAX). WORKING
C                        SPACE USED FOR THE SUBMATRICES ABOVE THE
C                        DIAGONAL OF THE U MATRIX OF THE LU-DECOMP.
C                       THE DIAGONAL SUBMATRICES OF U ARE UNIT 2*2.
C
C       ON RETURN
C
C
C               W        DOUBLE PRECISION(2,IXMAX).
C                        CONTAINS THE SOLUTION.
C
C***ROUTINES CALLED NONE
C***END PROLOGUE TRIDIA
C
C   Global variables.
C
      INTEGER IXMAX
      DOUBLE PRECISION A(2,2, - 1:1,IXMAX)
      DOUBLE PRECISION W(2,IXMAX),F(2,IXMAX),L(2,2, - 1:0,IXMAX)
      DOUBLE PRECISION U(2,2,1:1,IXMAX)
C
C   Local variables.
C
      INTEGER I,J
      DOUBLE PRECISION F1,F2,L21,U22
C
C       LU-DECOMPOSE.
C
C       FIRST STEP. (FIRST ROW)
C
C       L(0,1)=A(0,1)
C
C***FIRST EXECUTABLE STATEMENT TRIDIA
C
      DO 1 I = 1,2
          DO 1 J = 1,2
1     L(I,J,0,1) = A(I,J,0,1)
C
C       THE 2*2 SYSTEMS ARE SOLVED AS FOLLOWS:(AX=F)
C       L21=A21/A11
C       U22=A22-L21*A12
C       F2=F2-L21*F1
C       X2=F2/U22
C       X1=(F1-A12*X2)/A11
C
C       L(0,1)*U(1,1)=A(1,1)
C
      L21 = L(2,1,0,1)/L(1,1,0,1)
      U22 = L(2,2,0,1) - L21*L(1,2,0,1)
C       F2=A(2,1,1,1)-L21*A(1,1,1,1)(WE EXPLOIT ALL ZEROS)
      F2 = A(2,1,1,1)
      U(2,1,1,1) = F2/U22
C       U(1,1,1,1)=(A(1,1,1,1)-L(1,2,0,1)*U(2,1,1,1))/L(1,1,0,1)
      U(1,1,1,1) = - L(1,2,0,1)*U(2,1,1,1)/L(1,1,0,1)
C
      F1 = A(1,2,1,1)
      F2 = - L21*F1
      U(2,2,1,1) = F2/U22
      U(1,2,1,1) = (F1-L(1,2,0,1)*U(2,2,1,1))/L(1,1,0,1)
C
C       SECOND AND GENERAL STEP.
C
      DO 10 I = 2,IXMAX - 1
C
C       L(-1,I)=A(-1,I)
C
C         L(1,1,-1,I)=0
C         L(2,2,-1,I)=0
          L(1,2, - 1,I) = A(1,2, - 1,I)
          L(2,1, - 1,I) = A(2,1, - 1,I)
C
```

```
C      L(0,I)=A(0,I)-L(-1,I)*U(1,I-1)
C
         L(1,1,0,I) = A(1,1,0,I) - L(1,2, - 1,I)*U(2,1,1,I-1)
         L(1,2,0,I) = A(1,2,0,I) - L(1,2, - 1,I)*U(2,2,1,I-1)
         L(2,1,0,I) = A(2,1,0,I) - L(2,1, - 1,I)*U(1,1,1,I-1)
         L(2,2,0,I) = A(2,2,0,I) - L(2,1, - 1,I)*U(1,2,1,I-1)
C
C      L(0,I)*U(1,I)=A(1,I)
C
         L21 = L(2,1,0,I)/L(1,1,0,I)
         U22 = L(2,2,0,I) - L21*L(1,2,0,I)
         F2 = A(2,1,1,I)
         U(2,1,1,I) = F2/U22
         U(1,1,1,I) = - L(1,2,0,I)*U(2,1,1,I)/L(1,1,0,I)
         F1 = A(1,2,1,I)
         F2 = - L21*F1
         U(2,2,1,I) = F2/U22
         U(1,2,1,I) = (F1-L(1,2,0,I)*U(2,2,1,I))/L(1,1,0,I)
10    CONTINUE
C
C      END MAIN LOOP IN LU-DECOMP.
C
C      THIRD AND LAST STEP.(LAST ROW)
C
C      L(-1,IXMAX)=A(-1,IXMAX)
C
C      L(1,1,-1,IXMAX)=0
C      L(2,2,-1,IXMAX)=0
      L(1,2, - 1,IXMAX) = A(1,2, - 1,IXMAX)
C
C      L(0,IXMAX)=A(0,IXMAX)-L(-1,IXMAX)*U(1,IXMAX-1)
C
      L(1,1,0,IXMAX) = A(1,1,0,IXMAX) -
     +                 L(1,2, - 1,IXMAX)*U(2,1,1,IXMAX-1)
      L(1,2,0,IXMAX) = A(1,2,0,IXMAX) -
     +                 L(1,2, - 1,IXMAX)*U(2,2,1,IXMAX-1)
      L(2,1,0,IXMAX) = A(2,1,0,IXMAX) -
     +                 L(2,1, - 1,IXMAX)*U(1,1,1,IXMAX-1)
      L(2,2,0,IXMAX) = A(2,2,0,IXMAX) -
     +                 L(2,1, - 1,IXMAX)*U(1,2,1,IXMAX-1)
C
C      END LU FACTORISATION.
C
C      FORWARD SUBSTITUTION.
C
C      FIRST STEP.
C
      L21 = L(2,1,0,1)/L(1,1,0,1)
      U22 = L(2,2,0,1) - L21*L(1,2,0,1)
      F(2,1) = F(2,1) - L21*F(1,1)
      F(2,1) = F(2,1)/U22
      F(1,1) = (F(1,1)-L(1,2,0,1)*F(2,1))/L(1,1,0,1)
C
C      SECOND AND GENERAL STEP.
C
      DO 20 I = 2,IXMAX
         L21 = L(2,1,0,I)/L(1,1,0,I)
         U22 = L(2,2,0,I) - L21*L(1,2,0,I)
         F1 = F(1,I) - L(1,2, - 1,I)*F(2,I-1)
         F2 = F(2,I) - L(2,1, - 1,I)*F(1,I-1)
         F2 = F2 - L21*F1
         F(2,I) = F2/U22
         F(1,I) = (F1-L(1,2,0,I)*F(2,I))/L(1,1,0,I)
20    CONTINUE
C
C      END FORWARD SUBSTITUTION.
C
C      BACK SUBSTITUTION.
C
C      FIRST STEP.
C
      W(1,IXMAX) = F(1,IXMAX)
      W(2,IXMAX) = F(2,IXMAX)
C
C
C      SECOND AND GENERAL STEP.
C
      DO 30 I = IXMAX - 1,1, - 1
         W(1,I) = F(1,I) - U(1,1,1,I)*W(1,I+1)
     +            - U(1,2,1,I)*W(2,I+1)
         W(2,I) = F(2,I) - U(2,1,1,I)*W(1,I+1)
     +            - U(2,2,1,I)*W(2,I+1)
30    CONTINUE
C
C      END BACK SUBSTITUTION.
C
C***END TRIDIA
C
```

```
      RETURN
      END
```

incl.f

Fil som hører til `mbe_tefb.f`.

```
C**COMMON VARIABLES IN THE PROGRAM MBETBE
C
C   This file is a description of the global variables
C   used in connection with the
C   program mbetbe that simulates the sound field of a
C   spherical cap using the Mixed beam equation and the
C   Transformed beam equation
C
C   The variables are grouped according to their application
C
C***DATE WRITTEN   940412
C***AUTHOR
C            Bjoernar Ystad
C            Department of Mathematics,
C            University of Bergen,
C            Johannes Bruns gt. 12,
C            N-5007 Bergen, Norway
C
C**NOTE: Some of the variables are applied for several purposes, so
C        that if a variable is said to be necessary in one
C        application, it may be used in others as well.
C
C**Variables that are necessary in the routine widecyl
C
C         omega       double precision (0:ixmax,nharm,2), contains
C                     the boundary condition of the equation that
C                     is to be solved
C
C         ixmax       integer, is the number of inner points used
C                     in the numerics.
C
C         nharm       integer, is the number of harmonics that are
C                     retained in the calculations
C
C         mbflag      integer, equals 1 when the Mixed beam equation
C                     is solved, 2 when the Transformed beam
C                     equation is solved.
C
C         impstp      integer, is the number of fully implicit steps
C                     that are to be taken.
C
C         pi          double precision, is the mathematical constant
C
C         delta       double precision, is the distance between the
C                     origin of the Mixed beam equation and the Wide
C                     angle equation, measured in units of the focal
C                     length.
C
C         oangle      double precision, is the opening angle of the
C                     source.
C
C         pa          double precision
C         pd          double precision
C         pn          double precision, are numerical constants of
C                     the absorption- diffraction- and nonlinear
C                     terms respectively.
C
C         rstep       double precision, is the step length in the
C                     Richtmeyer procedure.
C
C         cc          double precision (0:ixmax), is the part of
C                     the diffraction coefficient that depends
C                     upon the transverse variable
C
C         cinvc       double precision (0:ixmax), is the part of
C                     the absorption coefficient that depends
C                     upon the transverse variable.
C
C         nonli       double precision (0:ixmax,nharm,2), contains
C                     the nonlinear contribution to the
C                     equations for the fourier coefficients.
C
C         cnew        double precision (0:ixmax,nharm), contains
C                     the absorption coefficient for the coming
C                     step (step no. j+1).
C
C         cold        double precision (0:ixmax,nharm), contains
C                     the absorption coefficient for the present
C                     step (step no. j).
C
C         khalf       double precision (0:ixmax,nharm), contains
C                     the diffraction coefficient.
C
C         uwidth      double precision, is the maximum value of the
C                     transverse variable of the Transformed beam
C                     equation.
C
C         ksimax      double precision, is the xi value of
C                     intersection between the xi axis and the
C                     line u = uwidth in the (zeta,xi)-plane.
C
C**Variables that are necessary in TRIDIA
C  The following documentation from TRIDIA defines the vaiables
C  transferred to this routine.
C
C      ON ENTRY
C              A       DOUBLE PRECISION(2,2,-1:1,IXMAX). DEFINES
C                      THE THREE-DIAGONAL MATRIX.
C                      THE FIRST TWO INDEXES DEFINES THE POSITION
C                      OF AN ELEMENT WITHIN A 2*2 SUB-MATRIX.
C                      THE THIRD INDEX DEFINES THE DISTANCE OF
C                      THE SUB-MATRIX FROM THE DIAGONAL.
C                      THE LAST INDEX DEFINES THE ROW NUMBER OF
C                      THE SUB-MATRIX.
C                      THIS INDEX CONVENTION IS ALSO USED FOR L
C                      AND U. UNCHANGED ON EXIT.
C
C              IXMAX   INTEGER. THE NUMBER OF DIAGONAL
C                      SUB-MATRICES.
C
C              F       DOUBLE PRECISION(2,IXMAX). DEFINES THE
C                      RIGHT-HAND SIDE. MODIFIED ON EXIT.
C
C              L       DOUBLE PRECISION(2,2,-1:0,IXMAX).
C                      WORKING SPACE
C                      USED FOR THE SUBMATRICES ON AND BELOW THE
C                      DIAGONAL
C                      OF THE L-MATRIX OF THE LU-DECOMPOSITION
C
C              U       DOUBLE PRECISION(2,2,1:1,IXMAX). WORKING
C                      SPACE USED FOR THE SUBMATRICES ABOVE THE
C                      DIAGONAL OF THE U MATRIX OF THE LU-DECOMP.
C                      THE DIAGONAL SUBMATRICES OF U ARE
C                      UNIT 2*2.
C
C      ON RETURN
C
C
C              W       DOUBLE PRECISION(2,IXMAX).
C                      CONTAINS THE SOLUTION.
C
C**Variables that are used in the interpolation
C
C          polgrid    double precision (0:ixmax,nharm,2,sigdim),
C                     contains data for the interpolation if the
C                     Mixed beam equation has been solved.
C
C          sigdim     integer, is the maximum number of
C                     solutions that can be stored in polgrid.
C
C          nstep      integer, is the number of solutions that
C                     have actually been solved.
C
C          firstsigma double precision, is the value of the
C                     evolution variable for the first solution
C                     that is solved in polgrid.
C
C          mx,my      integer, is the number of nodes in the x,y
C                     directions of the part of the plane in
C                     which the interpolation is performed
C
C          x          double precision (mx)
C          y          double precision (my), (x,y) are the
C                     coordinates of the  nodes in the zeta1,
C                     phimbe plane
C
C          tx,ty      double precision (ixmax+2), (tx,ty) are
C                     the points where we want to perform the
C                     interpolation
C
C          fi         double precision (mx*my), contains the
C                     function values of the nodes
C
C          ff         double precision (ixmax+2), contains the
C                     result of the interpolation
```

```
C
C**Variables necessary in the phase transformation.
C
C           kd         double precision, is the wave number times
C                      the focal length.
C
C           soangle    double precision, is sinus of the opening
C                      angle.
C
C**Variables that are necessary when generating plots of the
C  solution along the symmetry axis.
C
C        waxis      double precision (nant), contains the
C                   positions where axis data are to be written
C                   to file.
C
C        nant       integer, the number of points where axis data
C                   are to be registered.
C
C        acount     integer, is the number of points where axial
C                   data have been written to file.
C
C        filn       integer, is the ref. number of the file to
C                   which the axial data are written.
C
C        flength    double precision, is the focal length of the
C                   system
C
C        psource    double precision, is the reference pressure
C
CCCC variables for the Shapiro filter
C
C         shap       double precision (nharm), contains the
C                    "amplitude response function" of the
C                    Shapiro filter
C
C         ord_shap   integer; ord_shap+1 is the order
C                    of the Shapiro filter.
C
CCC variable used when reg. whether the Shapiro filter has
C   been used or not
C
C        logg_f_n     integer the unit number of the
C                     logg - file
C
CCC variable used when deciding whether the Shapiro filter is
C   to be used or not
C
C         max_energy   double precision contains the maximum
C                      on axis energy up to the present sigma
C                      step
C
C**integer variables
C
      integer ixmax,nharm,impstp,mbflag,desc_f_n,max,
     +        sigdim,mx,my,numbe,nstep,transv_f_n,
     +        nant,bnant,filn,acount,bcount,ord_shap,logg_f_n

C IXMAX VAR 1000
      parameter(ixmax=511,nharm=40,impstp=5,sigdim=64,
     +          numbe=100,mx=ixmax+2,my=sigdim,max=30,
     +          nant=500,bnant=2,filn=24,ord_shap=4)

      common/isc/ mbflag,nstep,logg_f_n,acount,bcount,
     +            transv_f_n,desc_f_n
C
C**double precision variables
C
      double precision oangle,pa,pn,kd,rstep,pd,pi,firstsigma,
     +                 flength,psource,tbedelta,dsigma,
     +                 soangle,uwidth,delta,max_energy,ksimax

      common/dsc/ oangle,rstep,pd,pi,firstsigma,
     +            soangle,delta,uwidth,max_energy,pa,pn,kd,flength,
     +            psource,ksimax,tbedelta,dsigma
C
C**integer arrays
C
      integer no_fc(bnant)

      common/iarr/ no_fc
C
C**double precision arrays
C
      double precision omega(0:ixmax,nharm,2),cc(0:ixmax),
     +                 cinvc(0:ixmax),waxis(nant),bwaxis(bnant),
     +                 nonli(0:ixmax,nharm,2),cnew(0:ixmax,nharm),
     +                 cold(0:ixmax,nharm),khalf(0:ixmax,nharm),
     +                 a(2,2,-1:1,0:ixmax),w(2,0:ixmax),
     +                 f(2,0:ixmax),work(12*ixmax+12),x(mx),
     +                 polgrid(0:ixmax,nharm,2,sigdim),
     +                 y(my),fi(mx,my),
     +                 tx(ixmax+2),
     +                 ty(ixmax+2),ff(ixmax+2),shap(nharm)

      common/darr/ omega,cc,cinvc,nonli,cnew,cold,khalf,a,w,f,work,
     +           polgrid,x,y,fi,tx,ty,ff,waxis,bwaxis,shap
```

itplbv.f

ACM-rutine for todimensjonal interpolasjon. Denne blir brukt i `tbe_maalt.f` og `mbe_tefb.f`.

```
C     ALGORITHM 474 COLLECTED ALGORITHMS FROM ACM.
C     ALGORITHM APPEARED IN COMM. ACM, VOL. VV, NO. NN,
C     P. 000.
      SUBROUTINE ITPLBV(IU, LX, LY, X, Y, Z, N, U, V, W)                A  10
C BIVARIATE INTERPOLATION
C THIS SUBROUTINE INTERPOLATES, FROM VALUES OF THE FUNCTION
C GIVEN AT INPUT GRID POINTS IN AN X-Y PLANE AND FOR A GIVEN
C SET OF POINTS IN THE PLANE, THE VALUES OF A SINGLE-VALUED
C BIVARIATE FUNCTION Z = Z(X,Y).
C THE METHOD IS BASED ON A PIECE-WISE FUNCTION COMPOSED OF
C A SET OF BICUBIC POLYNOMIALS IN X AND Y.  EACH POLYNOMIAL
C IS APPLICABLE TO A RECTANGLE OF THE INPUT GRID IN THE X-Y
C PLANE.  EACH POLYNOMIAL IS DETERMINED LOCALLY.
C THE INPUT PARAMETERS ARE
C IU  = LOGICAL UNIT NUMBER OF STANDARD OUTPUT UNIT
C LX  = NUMBER OF INPUT GRID POINTS IN THE X COORDINATE
C       (MUST BE 2 OR GREATER)
C LY  = NUMBER OF INPUT GRID POINTS IN THE Y COORDINATE
C       (MUST BE 2 OR GREATER)
C X   = ARRAY OF DIMENSION LX STORING THE X COORDINATES
C       OF INPUT GRID POINTS (IN ASCENDING ORDER)
C Y   = ARRAY OF DIMENSION LY STORING THE Y COORDINATES
C       OF INPUT GRID POINTS (IN ASCENDING ORDER)
C Z   = DOUBLY-DIMENSIONED ARRAY OF DIMENSION (LX,LY)
C       STORING THE VALUES OF THE FUNCTION (Z VALUES)
C       AT INPUT GRID POINTS
C N   = NUMBER OF POINTS AT WHICH INTERPOLATION OF THE
C       Z VALUE IS DESIRED (MUST BE 1 OR GREATER)
C U   = ARRAY OF DIMENSION N STORING THE X COORDINATES
C       OF DESIRED POINTS
C V   = ARRAY OF DIMENSION N STORING THE Y COORDINATES
C       OF DESIRED POINTS
C THE OUTPUT PARAMETER IS
C W   = ARRAY OF DIMENSION N WHERE THE INTERPOLATED Z
C       VALUES AT DESIRED POINTS ARE TO BE DISPLAYED
C       SOME VARIABLES INTERNALLY USED ARE
C ZA  = DIVIDED DIFFERENCE OF Z WITH RESPECT TO X
C ZB  = DIVIDED DIFFERENCE OF Z WITH RESPECT TO Y
C ZAB = SECOND ORDER DIVIDED DIFFERENCE OF Z WITH
C       RESPECT TO X AND Y
C ZX  = PARTIAL DERIVATIVE OF Z WITH RESPECT TO X
C ZY  = PARTIAL DERIVATIVE OF Z WITH RESPECT TO Y
C ZXY = SECOND ORDER PARTIAL DERIVATIVE OF Z WITH
C       RESPECT TO X AND Y
C DECLARATION STATEMENTS
      double precision x(lx),y(ly),z(lx,ly),u(n),v(n),w(n)
c      DIMENSION X(LX), Y(LY), Z(LX,LY), U(N), V(N), W(N)
      DIMENSION ZA(5,2), ZB(2,5), ZAB(3,3), ZX(4,4), ZY(4,4),
     * ZXY(4,4)
      EQUIVALENCE (Z3A1,ZA(1,1)), (Z3A2,ZA(2,1)), (Z3A3,ZA(3,1)),
     * (Z3A4,ZA(4,1)), (Z3A5,ZA(5,1)), (Z4A1,ZA(1,2)), (Z4A2,ZA(2,2)),
     * (Z4A3,ZA(3,2)), (Z4A4,ZA(4,2)), (Z4A5,ZA(5,2)), (Z3B1,ZB(1,1)),
     * (Z3B2,ZB(1,2)), (Z3B3,ZB(1,3)), (Z3B4,ZB(1,4)), (Z3B5,ZB(1,5)),
     * (Z4B1,ZB(2,1)), (Z4B2,ZB(2,2)), (Z4B3,ZB(2,3)), (Z4B4,ZB(2,4)),
     * (Z4B5,ZB(2,5)), (ZA2B2,ZAB(1,1)), (ZA3B2,ZAB(2,1)),
     * (ZA4B2,ZAB(3,1)), (ZA2B3,ZAB(1,2)), (ZA3B3,ZAB(2,2)),
     * (ZA4B3,ZAB(3,2)), (ZA2B4,ZAB(1,3)), (ZA3B4,ZAB(2,3)),
     * (ZA4B4,ZAB(3,3)), (ZX33,ZX(2,2)), (ZX43,ZX(3,2)),
     * (ZX34,ZX(2,3)), (ZX44,ZX(3,3)), (ZY33,ZY(2,2)),
     * (ZY43,ZY(3,2)), (ZY34,ZY(2,3)), (ZY44,ZY(3,3)),
     * (ZXY33,ZXY(2,2)), (ZXY43,ZXY(3,2)), (ZXY34,ZXY(2,3)),
     * (ZXY44,ZXY(3,3)), (P00,Z33), (P01,ZY33), (P10,ZX33),
     * (P11,ZXY33)
      EQUIVALENCE (LX0,ZX(1,1)), (LXM1,ZX(4,1)), (LXM2,ZX(1,4)),
     * (LXP1,ZX(4,4)), (LY0,ZY(1,1)), (LYM1,ZY(4,1)), (LYM2,ZY(1,4)),
     * (LYP1,ZY(4,4)), (IX,ZXY(1,1)), (IY,ZXY(4,1)), (IXPV,ZXY(1,4)),
     * (IYPV,ZXY(4,4)), (IMN,JX), (IMX,JY), (JXM2,JX1),
```

```
     * (JYM2,JY1), (UK,DX), (VK,DY), (A1,A5,B1,B5,ZX(2,1),A,Q0),
     * (A2,ZX(1,2),B,Q1), (A4,ZX(4,2),C,Q2), (B2,ZY(2,1),D,Q3),
     * (B4,ZY(2,4),E), (X2,ZX(3,1),A3SQ), (X4,ZX(1,3)), (X5,ZX(4,3)),
     * (Y2,ZX(2,4)), (Y4,ZY(3,1),B3SQ), (Y5,ZX(3,4),P02),
     * (Z23,ZY(1,2),P03), (Z24,ZY(4,2),P12), (Z32,ZY(1,3),P13),
     * (Z34,ZY(4,3),P20), (Z35,ZY(3,4),P21), (Z42,ZXY(2,1),P22),
     * (Z43,ZXY(1,2),P23), (Z44,ZXY(3,1),P30), (Z45,ZXY(4,2),P31),
     * (Z53,ZXY(1,3),P32), (Z54,ZXY(4,3),P33), (W2,WY2,W4),
     * (W3,WY3,W1,W5), (WX2,ZXY(2,4)), (WX3,ZXY(3,4))
C PRELIMINARY PROCESSING
C SETTING OF SOME INPUT PARAMETERS TO LOCAL VARIABLES
      IU0 = IU
      LX0 = LX
      LXM1 = LX0 - 1
      LXM2 = LXM1 - 1
      LXP1 = LX0 + 1
      LY0 = LY
      LYM1 = LY0 - 1
      LYM2 = LYM1 - 1
      LYP1 = LY0 + 1
      N0 = N
C ERROR CHECK
      IF (LXM2.LT.0) GO TO 710
      IF (LYM2.LT.0) GO TO 720
      IF (N0.LT.1) GO TO 730
      DO 10 IX=2,LX0
        IF (X(IX-1)-X(IX)) 10, 740, 750
   10 CONTINUE
      DO 20 IY=2,LY0
        IF (Y(IY-1)-Y(IY)) 20, 770, 780
   20 CONTINUE
C INITIAL SETTING OF PREVIOUS VALUES OF IX AND IY
      IXPV = 0
      IYPV = 0
C MAIN DO-LOOP
      DO 700 K=1,N0
        UK = U(K)
        VK = V(K)
C ROUTINES TO LOCATE THE DESIRED POINT
C TO FIND OUT THE IX VALUE FOR WHICH
C (U(K).GE.X(IX-1)).AND.(U(K).LT.X(IX))
        IF (LXM2.EQ.0) GO TO 80
        IF (UK.GE.X(LX0)) GO TO 70
        IF (UK.LT.X(1)) GO TO 60
        IMN = 2
        IMX = LX0
   30   IX = (IMN+IMX)/2
        IF (UK.GE.X(IX)) GO TO 40
        IMX = IX
        GO TO 50
   40   IMN = IX + 1
   50   IF (IMX.GT.IMN) GO TO 30
        IX = IMX
        GO TO 90
   60   IX = 1
        GO TO 90
   70   IX = LXP1
        GO TO 90
   80   IX = 2
C TO FIND OUT THE IY VALUE FOR WHICH
C (V(K).GE.Y(IY-1)).AND.(V(K).LT.Y(IY))
   90   IF (LYM2.EQ.0) GO TO 150
        IF (VK.GE.Y(LY0)) GO TO 140
        IF (VK.LT.Y(1)) GO TO 130
        IMN = 2
        IMX = LY0
  100   IY = (IMN+IMX)/2
        IF (VK.GE.Y(IY)) GO TO 110
        IMX = IY
        GO TO 120
  110   IMN = IY + 1
  120   IF (IMX.GT.IMN) GO TO 100
        IY = IMX
        GO TO 160
  130   IY = 1
        GO TO 160
  140   IY = LYP1
        GO TO 160
  150   IY = 2
C TO CHECK IF THE DESIRED POINT IS IN THE SAME RECTANGLE
C AS THE PREVIOUS POINT.  IF YES, SKIP TO THE COMPUTATION
C OF THE POLYNOMIAL
  160   IF (IX.EQ.IXPV .AND. IY.EQ.IYPV) GO TO 690
        IXPV = IX
        IYPV = IY
C ROUTINES TO PICK UP NECESSARY X, Y, AND Z VALUES, TO
C COMPUTE THE ZA, ZB, AND ZAB VALUES, AND TO ESTIMATE THEM
C WHEN NECESSARY
        JX = IX
        IF (JX.EQ.1) JX = 2
        IF (JX.EQ.LXP1) JX = LX0
        JY = IY
        IF (JY.EQ.1) JY = 2
        IF (JY.EQ.LYP1) JY = LY0
        JXM2 = JX - 2
        JXML = JX - LX0
        JYM2 = JY - 2
        JYML = JY - LY0
C IN THE CORE AREA, I.E., IN THE RECTANGLE THAT CONTAINS
C THE DESIRED POINT
        X3 = X(JX-1)
        X4 = X(JX)
        A3 = 1.0/(X4-X3)
        Y3 = Y(JY-1)
        Y4 = Y(JY)
        B3 = 1.0/(Y4-Y3)
        Z33 = Z(JX-1,JY-1)
        Z43 = Z(JX,JY-1)
        Z34 = Z(JX-1,JY)
        Z44 = Z(JX,JY)
        Z3A3 = (Z43-Z33)*A3
        Z4A3 = (Z44-Z34)*A3
        Z3B3 = (Z34-Z33)*B3
        Z4B3 = (Z44-Z43)*B3
        ZA3B3 = (Z4B3-Z3B3)*A3
C IN THE X DIRECTION
        IF (LXM2.EQ.0) GO TO 230
        IF (JXM2.EQ.0) GO TO 170
        X2 = X(JX-2)
        A2 = 1.0/(X3-X2)
        Z23 = Z(JX-2,JY-1)
        Z24 = Z(JX-2,JY)
        Z3A2 = (Z33-Z23)*A2
        Z4A2 = (Z34-Z24)*A2
        IF (JXML.EQ.0) GO TO 180
  170   X5 = X(JX+1)
        A4 = 1.0/(X5-X4)
        Z53 = Z(JX+1,JY-1)
        Z54 = Z(JX+1,JY)
        Z3A4 = (Z53-Z43)*A4
        Z4A4 = (Z54-Z44)*A4
        IF (JXM2.NE.0) GO TO 190
        Z3A2 = Z3A3 + Z3A3 - Z3A4
        Z4A2 = Z4A3 + Z4A3 - Z4A4
        GO TO 190
  180   Z3A4 = Z3A3 + Z3A3 - Z3A2
        Z4A4 = Z4A3 + Z4A3 - Z4A2
  190   ZA2B3 = (Z4A2-Z3A2)*B3
        ZA4B3 = (Z4A4-Z3A4)*B3
        IF (JX.LE.3) GO TO 200
        A1 = 1.0/(X2-X(JX-3))
        Z3A1 = (Z23-Z(JX-3,JY-1))*A1
        Z4A1 = (Z24-Z(JX-3,JY))*A1
        GO TO 210
  200   Z3A1 = Z3A2 + Z3A2 - Z3A3
        Z4A1 = Z4A2 + Z4A2 - Z4A3
  210   IF (JX.GE.LXM1) GO TO 220
        A5 = 1.0/(X(JX+2)-X5)
        Z3A5 = (Z(JX+2,JY-1)-Z53)*A5
        Z4A5 = (Z(JX+2,JY)-Z54)*A5
        GO TO 240
  220   Z3A5 = Z3A4 + Z3A4 - Z3A3
        Z4A5 = Z4A4 + Z4A4 - Z4A3
        GO TO 240
  230   Z3A2 = Z3A3
        Z4A2 = Z4A3
        GO TO 180
C IN THE Y DIRECTION
  240   IF (LYM2.EQ.0) GO TO 310
        IF (JYM2.EQ.0) GO TO 250
        Y2 = Y(JY-2)
        B2 = 1.0/(Y3-Y2)
        Z32 = Z(JX-1,JY-2)
        Z42 = Z(JX,JY-2)
        Z3B2 = (Z33-Z32)*B2
        Z4B2 = (Z43-Z42)*B2
        IF (JYML.EQ.0) GO TO 260
  250   Y5 = Y(JY+1)
        B4 = 1.0/(Y5-Y4)
        Z35 = Z(JX-1,JY+1)
        Z45 = Z(JX,JY+1)
        Z3B4 = (Z35-Z34)*B4
        Z4B4 = (Z45-Z44)*B4
        IF (JYM2.NE.0) GO TO 270
        Z3B2 = Z3B3 + Z3B3 - Z3B4
        Z4B2 = Z4B3 + Z4B3 - Z4B4
```

```
      GO TO 270
  260 Z3B4 = Z3B3 + Z3B3 - Z3B2
      Z4B4 = Z4B3 + Z4B3 - Z4B2
  270 ZA3B2 = (Z4B2-Z3B2)*A3
      ZA3B4 = (Z4B4-Z3B4)*A3
      IF (JY.LE.3) GO TO 280
      B1 = 1.0/(Y2-Y(JY-3))
      Z3B1 = (Z32-Z(JX-1,JY-3))*B1
      Z4B1 = (Z42-Z(JX,JY-3))*B1
      GO TO 290
  280 Z3B1 = Z3B2 + Z3B2 - Z3B3
      Z4B1 = Z4B2 + Z4B2 - Z4B3
  290 IF (JY.GE.LYM1) GO TO 300
      B5 = 1.0/(Y(JY+2)-Y5)
      Z3B5 = (Z(JX-1,JY+2)-Z35)*B5
      Z4B5 = (Z(JX,JY+2)-Z45)*B5
      GO TO 320
  300 Z3B5 = Z3B4 + Z3B4 - Z3B3
      Z4B5 = Z4B4 + Z4B4 - Z4B3
      GO TO 320
  310 Z3B2 = Z3B3
      Z4B2 = Z4B3
      GO TO 260
C IN THE DIAGONAL DIRECTIONS
  320 IF (LXM2.EQ.0) GO TO 400
      IF (LYM2.EQ.0) GO TO 410
      IF (JXML.EQ.0) GO TO 350
      IF (JYM2.EQ.0) GO TO 330
      ZA4B2 = ((Z53-Z(JX+1,JY-2))*B2-Z4B2)*A4
      IF (JYML.EQ.0) GO TO 340
  330 ZA4B4 = ((Z(JX+1,JY+1)-Z54)*B4-Z4B4)*A4
      IF (JYM2.NE.0) GO TO 380
      ZA4B2 = ZA4B3 + ZA4B3 - ZA4B4
      GO TO 380
  340 ZA4B4 = ZA4B3 + ZA4B3 - ZA4B2
      GO TO 380
  350 IF (JYM2.EQ.0) GO TO 360
      ZA2B2 = (Z3B2-(Z23-Z(JX-2,JY-2))*B2)*A2
      IF (JYML.EQ.0) GO TO 370
  360 ZA2B4 = (Z3B4-(Z(JX-2,JY+1)-Z24)*B4)*A2
      IF (JYM2.NE.0) GO TO 390
      ZA2B2 = ZA2B3 + ZA2B3 - ZA2B4
      GO TO 390
  370 ZA2B4 = ZA2B3 + ZA2B3 - ZA2B2
      GO TO 390
  380 IF (JXM2.NE.0) GO TO 350
      ZA2B2 = ZA3B2 + ZA3B2 - ZA4B2
      ZA2B4 = ZA3B4 + ZA3B4 - ZA4B4
      GO TO 420
  390 IF (JXML.NE.0) GO TO 420
      ZA4B2 = ZA3B2 + ZA3B2 - ZA2B2
      ZA4B4 = ZA3B4 + ZA3B4 - ZA2B4
      GO TO 420
  400 ZA2B2 = ZA3B2
      ZA4B2 = ZA3B2
      ZA2B4 = ZA3B4
      ZA4B4 = ZA3B4
      GO TO 420
  410 ZA2B2 = ZA2B3
      ZA2B4 = ZA2B3
      ZA4B2 = ZA4B3
      ZA4B4 = ZA4B3
C NUMERICAL DIFFERENTIATION   ---   TO DETERMINE PARTIAL
C DERIVATIVES ZX, ZY, AND ZXY AS WEIGHTED MEANS OF DIVIDED
C DIFFERENCES ZA, ZB, AND ZAB, RESPECTIVELY
  420 DO 480 JY=2,3
        DO 470 JX=2,3
          W2 = ABS(ZA(JX+2,JY-1)-ZA(JX+1,JY-1))
          W3 = ABS(ZA(JX,JY-1)-ZA(JX-1,JY-1))
          SW = W2 + W3
          IF (SW.EQ.0.0) GO TO 430
          WX2 = W2/SW
          WX3 = W3/SW
          GO TO 440
  430     WX2 = 0.5
          WX3 = 0.5
  440     ZX(JX,JY) = WX2*ZA(JX,JY-1) + WX3*ZA(JX+1,JY-1)
          W2 = ABS(ZB(JX-1,JY+2)-ZB(JX-1,JY+1))
          W3 = ABS(ZB(JX-1,JY)-ZB(JX-1,JY-1))
          SW = W2 + W3
          IF (SW.EQ.0.0) GO TO 450
          WY2 = W2/SW
          WY3 = W3/SW
          GO TO 460
  450     WY2 = 0.5
          WY3 = 0.5
  460     ZY(JX,JY) = WY2*ZB(JX-1,JY) + WY3*ZB(JX-1,JY+1)
          ZXY(JX,JY) =
     *      WY2*(WX2*ZAB(JX-1,JY-1)+WX3*ZAB(JX,JY-1)) +
     *      WY3*(WX2*ZAB(JX-1,JY)+WX3*ZAB(JX,JY))
  470   CONTINUE
  480 CONTINUE
C WHEN (U(K).LT.X(1)).OR.(U(K).GT.X(LX))
      IF (IX.EQ.LXP1) GO TO 530
      IF (IX.NE.1) GO TO 590
      W2 = A4*(3.0*A3+A4)
      W1 = 2.0*A3*(A3-A4) + W2
      DO 500 JY=2,3
        ZX(1,JY) = (W1*ZA(1,JY-1)+W2*ZA(2,JY-1))/(W1+W2)
        ZY(1,JY) = ZY(2,JY) + ZY(2,JY) - ZY(3,JY)
        ZXY(1,JY) = ZXY(2,JY) + ZXY(2,JY) - ZXY(3,JY)
        DO 490 JX1=2,3
          JX = 5 - JX1
          ZX(JX,JY) = ZX(JX-1,JY)
          ZY(JX,JY) = ZY(JX-1,JY)
          ZXY(JX,JY) = ZXY(JX-1,JY)
  490   CONTINUE
  500 CONTINUE
      X3 = X3 - 1.0/A4
      Z33 = Z33 - Z3A2/A4
      DO 510 JY=1,5
        ZB(2,JY) = ZB(1,JY)
  510 CONTINUE
      DO 520 JY=2,4
        ZB(1,JY) = ZB(1,JY) - ZAB(1,JY-1)/A4
  520 CONTINUE
      A3 = A4
      JX = 1
      GO TO 570
  530 W4 = A2*(3.0*A3+A2)
      W5 = 2.0*A3*(A3-A2) + W4
      DO 550 JY=2,3
        ZX(4,JY) = (W4*ZA(4,JY-1)+W5*ZA(5,JY-1))/(W4+W5)
        ZY(4,JY) = ZY(3,JY) + ZY(3,JY) - ZY(2,JY)
        ZXY(4,JY) = ZXY(3,JY) + ZXY(3,JY) - ZXY(2,JY)
        DO 540 JX=2,3
          ZX(JX,JY) = ZX(JX+1,JY)
          ZY(JX,JY) = ZY(JX+1,JY)
          ZXY(JX,JY) = ZXY(JX+1,JY)
  540   CONTINUE
  550 CONTINUE
      X3 = X4
      Z33 = Z43
      DO 560 JY=1,5
        ZB(1,JY) = ZB(2,JY)
  560 CONTINUE
      A3 = A2
      JX = 3
  570 ZA(3,1) = ZA(JX+1,1)
      DO 580 JY=1,3
        ZAB(2,JY) = ZAB(JX,JY)
  580 CONTINUE
C WHEN (V(K).LT.Y(1)).OR.(V(K).GT.Y(LY))
  590 IF (IY.EQ.LYP1) GO TO 630
      IF (IY.NE.1) GO TO 680
      W2 = B4*(3.0*B3+B4)
      W1 = 2.0*B3*(B3-B4) + W2
      DO 620 JX=2,3
        IF (JX.EQ.3 .AND. IX.EQ.LXP1) GO TO 600
        IF (JX.EQ.2 .AND. IX.EQ.1) GO TO 600
        ZY(JX,1) = (W1*ZB(JX-1,1)+W2*ZB(JX-1,2))/(W1+W2)
        ZX(JX,1) = ZX(JX,2) + ZX(JX,2) - ZX(JX,3)
        ZXY(JX,1) = ZXY(JX,2) + ZXY(JX,2) - ZXY(JX,3)
  600   DO 610 JY1=2,3
          JY = 5 - JY1
          ZY(JX,JY) = ZY(JX,JY-1)
          ZX(JX,JY) = ZX(JX,JY-1)
          ZXY(JX,JY) = ZXY(JX,JY-1)
  610   CONTINUE
  620 CONTINUE
      Y3 = Y3 - 1.0/B4
      Z33 = Z33 - Z3B2/B4
      Z3A3 = Z3A3 - ZA3B2/B4
      Z3B3 = Z3B2
      ZA3B3 = ZA3B2
      B3 = B4
      GO TO 670
  630 W4 = B2*(3.0*B3+B2)
      W5 = 2.0*B3*(B3-B2) + W4
      DO 660 JX=2,3
        IF (JX.EQ.3 .AND. IX.EQ.LXP1) GO TO 640
        IF (JX.EQ.2 .AND. IX.EQ.1) GO TO 640
        ZY(JX,4) = (W4*ZB(JX-1,4)+W5*ZB(JX-1,5))/(W4+W5)
        ZX(JX,4) = ZX(JX,3) + ZX(JX,3) - ZX(JX,2)
        ZXY(JX,4) = ZXY(JX,3) + ZXY(JX,3) - ZXY(JX,2)
  640   DO 650 JY=2,3
```

```
          ZY(JX,JY) = ZY(JX,JY+1)
          ZX(JX,JY) = ZX(JX,JY+1)
          ZXY(JX,JY) = ZXY(JX,JY+1)
  650   CONTINUE
  660   CONTINUE
        Y3 = Y4
        Z33 = Z33 + Z3B3/B3
        Z3A3 = Z3A3 + ZA3B3/B3
        Z3B3 = Z3B4
        ZA3B3 = ZA3B4
        B3 = B2
  670   IF (IX.NE.1 .AND. IX.NE.LXP1) GO TO 680
        JX = IX/LXP1 + 2
        JX1 = 5 - JX
        JY = IY/LYP1 + 2
        JY1 = 5 - JY
        ZX(JX,JY) = ZX(JX1,JY) + ZX(JX,JY1) - ZX(JX1,JY1)
        ZY(JX,JY) = ZY(JX1,JY) + ZY(JX,JY1) - ZY(JX1,JY1)
        ZXY(JX,JY) = ZXY(JX1,JY) + ZXY(JX,JY1) - ZXY(JX1,JY1)
C DETERMINATION OF THE COEFFICIENTS OF THE POLYNOMIAL
  680   ZX3B3 = (ZX34-ZX33)*B3
        ZX4B3 = (ZX44-ZX43)*B3
        ZY3A3 = (ZY43-ZY33)*A3
        ZY4A3 = (ZY44-ZY34)*A3
        A = ZA3B3 - ZX3B3 - ZY3A3 + ZXY33
        B = ZX4B3 - ZX3B3 - ZXY43 + ZXY33
        C = ZY4A3 - ZY3A3 - ZXY34 + ZXY33
        D = ZXY44 - ZXY43 - ZXY34 + ZXY33
        E = A + A - B - C
        A3SQ = A3*A3
        B3SQ = B3*B3
        P02 = (2.0*(Z3B3-ZY33)+Z3B3-ZY34)*B3
        P03 = (-2.0*Z3B3+ZY34+ZY33)*B3SQ
        P12 = (2.0*(ZX3B3-ZXY33)+ZX3B3-ZXY34)*B3
        P13 = (-2.0*ZX3B3+ZXY34+ZXY33)*B3SQ
        P20 = (2.0*(Z3A3-ZX33)+Z3A3-ZX43)*A3
        P21 = (2.0*(ZY3A3-ZXY33)+ZY3A3-ZXY43)*A3
        P22 = (3.0*(A+E)+D)*A3*B3
        P23 = (-3.0*E-B-D)*A3*B3SQ
        P30 = (-2.0*Z3A3+ZX43+ZX33)*A3SQ
        P31 = (-2.0*ZY3A3+ZXY43+ZXY33)*A3SQ
        P32 = (-3.0*E-C-D)*B3*A3SQ
        P33 = (D+E+E)*A3SQ*B3SQ
C COMPUTATION OF THE POLYNOMIAL
  690   DY = VK - Y3
        Q0 = P00 + DY*(P01+DY*(P02+DY*P03))
        Q1 = P10 + DY*(P11+DY*(P12+DY*P13))
        Q2 = P20 + DY*(P21+DY*(P22+DY*P23))
        Q3 = P30 + DY*(P31+DY*(P32+DY*P33))
        DX = UK - X3
        W(K) = Q0 + DX*(Q1+DX*(Q2+DX*Q3))
  700 CONTINUE
C NORMAL EXIT
      RETURN
C ERROR EXIT
  710 WRITE (IU0,99999)
      GO TO 800
  720 WRITE (IU0,99998)
      GO TO 800
  730 WRITE (IU0,99997)
      GO TO 800
  740 WRITE (IU0,99996)
      GO TO 760
  750 WRITE (IU0,99995)
  760 WRITE (IU0,99994) IX, X(IX)
      GO TO 800
  770 WRITE (IU0,99993)
      GO TO 790
  780 WRITE (IU0,99992)
  790 WRITE (IU0,99991) IY, Y(IY)
  800 WRITE (IU0,99990) LX0, LY0, N0
      RETURN
C FORMAT STATEMENTS
99999 FORMAT(1X/23H  ***   LX = 1 OR LESS./)
99998 FORMAT(1X/23H  ***   LY = 1 OR LESS./)
99997 FORMAT(1X/22H  ***   N = 0 OR LESS./)
99996 FORMAT(1X/27H  ***   IDENTICAL X VALUES./)
99995 FORMAT(1X/33H  ***   X VALUES OUT OF SEQUENCE./)
99994 FORMAT(7H   IX =, I6, 10X, 7HX(IX) =, E12.3)
99993 FORMAT(1X/27H  ***   IDENTICAL Y VALUES./)
99992 FORMAT(1X/33H  ***   Y VALUES OUT OF SEQUENCE./)
99991 FORMAT(7H   IY =, I6, 10X, 7HY(IY) =, E12.3)
99990 FORMAT(7H   LX =, I6, 10X, 4HLY =, I6, 10X, 3HN =, I7/
     * 36H ERROR DETECTED IN ROUTINE    ITPLBV)
      END
      SUBROUTINE SFCFIT(IU, LX, LY, X, Y, Z, MX, MY, NU, NV, U,
     * V, W)
C SMOOTH SURFACE FITTING
C THIS SUBROUTINE FITS A SMOOTH SURFACE OF A SINGLE-VALUED
C BIVARIATE FUNCTION Z = Z(X,Y) TO A SET OF INPUT DATA
C POINTS GIVEN AT INPUT GRID POINTS IN AN X-Y PLANE.  IT
C GENERATES A SET OF OUTPUT GRID POINTS BY EQUALLY DIVIDING
C THE X AND Y COORDINATES IN EACH INTERVAL BETWEEN A PAIR
C OF INPUT GRID POINTS, INTERPOLATES THE Z VALUE FOR THE
C X AND Y VALUES OF EACH OUTPUT GRID POINT, AND GENERATES
C A SET OF OUTPUT POINTS CONSISTING OF INPUT DATA POINTS
C AND THE INTERPOLATED POINTS.
C THE METHOD IS BASED ON A PIECE-WISE FUNCTION COMPOSED OF
C A SET OF BICUBIC POLYNOMIALS IN X AND Y.  EACH POLYNOMIAL
C IS APPLICABLE TO A RECTANGLE OF THE INPUT GRID IN THE X-Y
C PLANE.  EACH POLYNOMIAL IS DETERMINED LOCALLY.
C THE INPUT PARAMETERS ARE
C IU  = LOGICAL UNIT NUMBER OF STANDARD OUTPUT UNIT
C LX  = NUMBER OF INPUT GRID POINTS IN THE X COORDINATE
C       (MUST BE 2 OR GREATER)
C LY  = NUMBER OF INPUT GRID POINTS IN THE Y COORDINATE
C       (MUST BE 2 OR GREATER)
C X   = ARRAY OF DIMENSION LX STORING THE X COORDINATES
C       OF INPUT GRID POINTS (IN ASCENDING OR DESCENDING
C       ORDER)
C Y   = ARRAY OF DIMENSION LY STORING THE Y COORDINATES
C       OF INPUT GRID POINTS (IN ASCENDING OR DESCENDING
C       ORDER)
C Z   = DOUBLY-DIMENSIONED ARRAY OF DIMENSION (LX,LY)
C       STORING THE VALUES OF THE FUNCTION AT INPUT
C       GRID POINTS
C MX  = NUMBER OF SUBINTERVALS BETWEEN EACH PAIR OF
C       INPUT GRID POINTS IN THE X COORDINATE
C       (MUST BE 2 OR GREATER)
C MY  = NUMBER OF SUBINTERVALS BETWEEN EACH PAIR OF
C       INPUT GRID POINTS IN THE Y COORDINATE
C       (MUST BE 2 OR GREATER)
C NU  = NUMBER OF OUTPUT GRID POINTS IN THE X COORDINATE
C     = (LX-1)*MX+1
C NV  = NUMBER OF OUTPUT GRID POINTS IN THE Y COORDINATE
C     = (LY-1)*MY+1
C THE OUTPUT PARAMETERS ARE
C U   = ARRAY OF DIMENSION NU WHERE THE X COORDINATES OF
C       OUTPUT POINTS ARE TO BE DISPLAYED
C V   = ARRAY OF DIMENSION NV WHERE THE Y COORDINATES OF
C       OUTPUT POINTS ARE TO BE DISPLAYED
C W   = DOUBLY-DIMENSIONED ARRAY OF DIMENSION (NU,NV)
C       WHERE THE Z COORDINATES OF OUTPUT POINTS ARE TO
C       BE DISPLAYED
C SOME VARIABLES INTERNALLY USED ARE
C ZA  = DIVIDED DIFFERENCE OF Z WITH RESPECT TO X
C ZB  = DIVIDED DIFFERENCE OF Z WITH RESPECT TO Y
C ZAB = SECOND ORDER DIVIDED DIFFERENCE OF Z WITH
C       RESPECT TO X AND Y
C ZX  = PARTIAL DERIVATIVE OF Z WITH RESPECT TO X
C ZY  = PARTIAL DERIVATIVE OF Z WITH RESPECT TO Y
C ZXY = SECOND ORDER PARTIAL DERIVATIVE OF Z WITH
C       RESPECT TO X AND Y
C DECLARATION STATEMENTS
      DIMENSION X(LX), Y(LY), Z(LX,LY), U(NU), V(NV), W(NU,NV)
      DIMENSION ZA(4,2), ZB(5), ZAB(2,3), ZX(2), ZY(2), ZXY(2)
      EQUIVALENCE (Z3A2,ZA(1,1)), (Z3A3,ZA(2,1)), (Z3A4,ZA(3,1)),
     * (Z3A5,ZA(4,1)), (Z4A2,ZA(1,2)), (Z4A3,ZA(2,2)), (Z4A4,ZA(3,2)),
     * (Z4A5,ZA(4,2)), (Z4B1,ZB(1)), (Z4B2,ZB(2)), (Z4B3,ZB(3)),
     * (Z4B4,ZB(4)), (Z4B5,ZB(5)), (ZA3B2,ZAB(1,1)),
     * (ZA4B2,ZAB(2,1)), (ZA3B3,ZAB(1,2)), (ZA4B3,ZAB(2,2)),
     * (ZA3B4,ZAB(1,3)), (ZA4B4,ZAB(2,3)), (ZX43,ZX(1)),
     * (ZX44,ZX(2)), (ZY43,ZY(1)), (ZY44,ZY(2)),
     * (ZXY43,ZXY(1)), (ZXY44,ZXY(2)), (P00,Z33), (P01,ZY33),
     * (P10,ZX33), (P11,ZXY33)
      EQUIVALENCE (IXM1,JX), (IXML,JY), (DU,DV,DX,DY),
     * (FMX,RMX,FMY,RMY,SW,E), (W2,WY2,A,Q0), (W3,WY3,B,Q1),
     * (WX2,C,Q2), (WX3,D,Q3), (Z3A2,P02), (Z4A2,P03),
     * (Z4B1,P12), (Z4B2,P13), (Z4B4,P20), (Z4B5,P21),
     * (ZA3B2,P22), (ZA3B4,P23)
C PRELIMINARY PROCESSING
C SETTING OF SOME INPUT PARAMETERS TO LOCAL VARIABLES
      IU0 = IU
      LX0 = LX
      LXM1 = LX0 - 1
      LXM2 = LXM1 - 1
      LY0 = LY
      LYM1 = LY0 - 1
      LYM2 = LYM1 - 1
      MX0 = MX
      MXP1 = MX0 + 1
      MXM1 = MX0 - 1
      MY0 = MY
      MYP1 = MY0 + 1
      MYM1 = MY0 - 1
      NU0 = NU
```

```
      NVO = NV
C ERROR CHECK
      IF (LXM2.LT.0) GO TO 400
      IF (LYM2.LT.0) GO TO 410
      IF (MXM1.LE.0) GO TO 420
      IF (MYM1.LE.0) GO TO 430
      IF (NUO.NE.LXM1*MXO+1) GO TO 440
      IF (NVO.NE.LYM1*MYO+1) GO TO 450
      IX = 2
      IF (X(1)-X(2)) 10, 460, 30
   10 DO 20 IX=3,LXO
        IF (X(IX-1)-X(IX)) 20, 460, 470
   20 CONTINUE
      GO TO 50
   30 DO 40 IX=3,LXO
        IF (X(IX-1)-X(IX)) 470, 460, 40
   40 CONTINUE
   50 IY = 2
      IF (Y(1)-Y(2)) 60, 490, 80
   60 DO 70 IY=3,LYO
        IF (Y(IY-1)-Y(IY)) 70, 490, 500
   70 CONTINUE
      GO TO 100
   80 DO 90 IY=3,LYO
        IF (Y(IY-1)-Y(IY)) 500, 490, 90
   90 CONTINUE
C COMPUTATION OF THE U ARRAY
  100 FMX = MXO
      RMX = 1.0/FMX
      KU = 1
      X4 = X(1)
      U(1) = X4
      DO 120 IX=2,LXO
        X3 = X4
        X4 = X(IX)
        DU = (X4-X3)*RMX
        DO 110 JX=1,MXM1
          KU = KU + 1
          U(KU) = U(KU-1) + DU
  110   CONTINUE
        KU = KU + 1
        U(KU) = X4
  120 CONTINUE
C COMPUTATION OF THE V ARRAY
      FMY = MYO
      RMY = 1.0/FMY
      KV = 1
      Y4 = Y(1)
      V(1) = Y4
      DO 140 IY=2,LYO
        Y3 = Y4
        Y4 = Y(IY)
        DV = (Y4-Y3)*RMY
        DO 130 JY=1,MYM1
          KV = KV + 1
          V(KV) = V(KV-1) + DV
  130   CONTINUE
        KV = KV + 1
        V(KV) = Y4
  140 CONTINUE
C MAIN DO-LOOPS
      JYMX = MYO
      KVO = 0
      DO 390 IY=2,LYO
        IYM2 = IY - 2
        IYM3 = IYM2 - 1
        IYML = IY - LYO
        IYML1 = IYML + 1
        IX6 = 0
        IF (IYML.EQ.0) JYMX = MYP1
        JXMX = MXO
        KUO = 0
        DO 380 IX=1,LXO
          IXM1 = IX - 1
          IXML = IX - LXO
          IF (IXML.EQ.0) JXMX = MXP1
C ROUTINES TO PICK UP NECESSARY X, Y, AND Z VALUES, TO
C COMPUTE THE ZA, ZB, AND ZAB VALUES, AND TO ESTIMATE THEM
C WHEN NECESSARY
C PRELIMINARY WHEN IX.EQ.1
          IF (IXM1.NE.0) GO TO 150
          Y3 = Y(IY-1)
          Y4 = Y(IY)
          B3 = 1.0/(Y4-Y3)
          B3SQ = B3*B3
          IF (IYM2.GT.0) B2 = 1.0/(Y3-Y(IY-2))
          IF (IYM3.GT.0) B1 = 1.0/(Y(IY-2)-Y(IY-3))
          IF (IYML.LT.0) B4 = 1.0/(Y(IY+1)-Y4)
          IF (IYML1.LT.0) B5 = 1.0/(Y(IY+2)-Y(IY+1))
          GO TO 180
C TO SAVE THE OLD VALUES
  150     Z3A2 = Z3A3
          Z4A2 = Z4A3
          X3 = X4
          Z33 = Z43
          Z3B3 = Z4B3
          A3 = A4
          A3SQ = A3*A3
          Z3A3 = Z3A4
          Z4A3 = Z4A4
          ZA3B2 = ZA4B2
          ZA3B3 = ZA4B3
          ZA3B4 = ZA4B4
  160     X4 = X5
          Z43 = Z53
          Z4B1 = Z5B1
          Z4B2 = Z5B2
          Z4B3 = Z5B3
          Z4B4 = Z5B4
          Z4B5 = Z5B5
          A4 = A5
          Z3A4 = Z3A5
          Z4A4 = Z4A5
          ZA4B2 = ZA5B2
          ZA4B3 = ZA5B3
          ZA4B4 = ZA5B4
  170     X5 = X6
          Z53 = Z63
          Z54 = Z64
          Z5B1 = Z6B1
          Z5B2 = Z6B2
          Z5B3 = Z6B3
          Z5B4 = Z6B4
          Z5B5 = Z6B5
C TO COMPUTE THE ZA, ZB, AND ZAB VALUES AND
C TO ESTIMATE THE ZB VALUES
C WHEN (IY.LE.3).OR.(IY.GE.LY-1)
  180     IX6 = IX6 + 1
          IF (IX6.GT.LXO) GO TO 260
          X6 = X(IX6)
          Z63 = Z(IX6,IY-1)
          Z64 = Z(IX6,IY)
          Z6B3 = (Z64-Z63)*B3
          IF (LYM2.EQ.0) GO TO 200
          IF (IYM2.EQ.0) GO TO 190
          Z62 = Z(IX6,IY-2)
          Z6B2 = (Z63-Z62)*B2
          IF (IYML.NE.0) GO TO 190
          Z6B4 = Z6B3 + Z6B3 - Z6B2
          GO TO 210
  190     Z65 = Z(IX6,IY+1)
          Z6B4 = (Z65-Z64)*B4
          IF (IYM2.NE.0) GO TO 210
          Z6B2 = Z6B3 + Z6B3 - Z6B4
          GO TO 210
  200     Z6B2 = Z6B3
          Z6B4 = Z6B3
  210     IF (IYM3.LE.0) GO TO 220
          Z6B1 = (Z62-Z(IX6,IY-3))*B1
          GO TO 230
  220     Z6B1 = Z6B2 + Z6B2 - Z6B3
  230     IF (IYML1.GE.0) GO TO 240
          Z6B5 = (Z(IX6,IY+2)-Z65)*B5
          GO TO 250
  240     Z6B5 = Z6B4 + Z6B4 - Z6B3
  250     IF (IX6.EQ.1) GO TO 170
          A5 = 1.0/(X6-X5)
          Z3A5 = (Z63-Z53)*A5
          Z4A5 = (Z64-Z54)*A5
          ZA5B2 = (Z6B2-Z5B2)*A5
          ZA5B3 = (Z6B3-Z5B3)*A5
          ZA5B4 = (Z6B4-Z5B4)*A5
          IF (IX6.EQ.2) GO TO 160
          GO TO 280
C TO ESTIMATE THE ZA AND ZAB VALUES
C WHEN (IX.GE.LX-1).AND.(LX.GT.2)
  260     IF (LXM2.EQ.0) GO TO 270
          Z3A5 = Z3A4 + Z3A4 - Z3A3
          Z4A5 = Z4A4 + Z4A4 - Z4A3
          IF (IXML.EQ.0) GO TO 290
          ZA5B2 = ZA4B2 + ZA4B2 - ZA3B2
          ZA5B3 = ZA4B3 + ZA4B3 - ZA3B3
          ZA5B4 = ZA4B4 + ZA4B4 - ZA3B4
          GO TO 290
C TO ESTIMATE THE ZA AND ZAB VALUES
C WHEN (IX.GE.LX-1).AND.(LX.EQ.2)
```

```
  270     Z3A5 = Z3A4
          Z4A5 = Z4A4
          IF (IXML.EQ.0) GO TO 290
          ZA5B2 = ZA4B2
          ZA5B3 = ZA4B3
          ZA5B4 = ZA4B4
C TO ESTIMATE THE ZA AND ZAB VALUES
C WHEN IX.EQ.1
  280     IF (IXM1.NE.0) GO TO 290
          Z3A3 = Z3A4 + Z3A4 - Z3A5
          Z3A2 = Z3A3 + Z3A3 - Z3A4
          Z4A3 = Z4A4 + Z4A4 - Z4A5
          Z4A2 = Z4A3 + Z4A3 - Z4A4
          ZA3B2 = ZA4B2 + ZA4B2 - ZA5B2
          ZA3B3 = ZA4B3 + ZA4B3 - ZA5B3
          ZA3B4 = ZA4B4 + ZA4B4 - ZA5B4
          GO TO 300
C NUMERICAL DIFFERENTIATION   ---   TO DETERMINE PARTIAL
C DERIVATIVES ZX, ZY, AND ZXY AS WEIGHTED MEANS OF DIVIDED
C DIFFERENCES ZA, ZB, AND ZAB, RESPECTIVELY
C TO SAVE THE OLD VALUES WHEN IX.NE.1
  290     ZX33 = ZX43
          ZX34 = ZX44
          ZY33 = ZY43
          ZY34 = ZY44
          ZXY33 = ZXY43
          ZXY34 = ZXY44
C NEW COMPUTATION
  300     DO 350 JY=1,2
            W2 = ABS(ZA(4,JY)-ZA(3,JY))
            W3 = ABS(ZA(2,JY)-ZA(1,JY))
            SW = W2 + W3
            IF (SW.EQ.0.0) GO TO 310
            WX2 = W2/SW
            WX3 = W3/SW
            GO TO 320
  310       WX2 = 0.5
            WX3 = 0.5
  320       ZX(JY) = WX2*ZA(2,JY) + WX3*ZA(3,JY)
            W2 = ABS(ZB(JY+3)-ZB(JY+2))
            W3 = ABS(ZB(JY+1)-ZB(JY))
            SW = W2 + W3
            IF (SW.EQ.0.0) GO TO 330
            WY2 = W2/SW
            WY3 = W3/SW
            GO TO 340
  330       WY2 = 0.5
            WY3 = 0.5
  340       ZY(JY) = WY2*ZB(JY+1) + WY3*ZB(JY+2)
            ZXY(JY) = WY2*(WX2*ZAB(1,JY)+WX3*ZAB(2,JY)) +
     *      WY3*(WX2*ZAB(1,JY+1)+WX3*ZAB(2,JY+1))
  350     CONTINUE
          IF (IXM1.EQ.0) GO TO 380
C DETERMINATION OF THE COEFFICIENTS OF THE POLYNOMIAL
          ZX3B3 = (ZX34-ZX33)*B3
          ZX4B3 = (ZX44-ZX43)*B3
          ZY3A3 = (ZY43-ZY33)*A3
          ZY4A3 = (ZY44-ZY34)*A3
          A = ZA3B3 - ZX3B3 - ZY3A3 + ZXY33
          B = ZX4B3 - ZX3B3 - ZXY43 + ZXY33
          C = ZY4A3 - ZY3A3 - ZXY34 + ZXY33
          D = ZXY44 - ZXY43 - ZXY34 + ZXY33
          E = A + A - B - C
          P02 = (2.0*(Z3B3-ZY33)+Z3B3-ZY34)*B3
          P03 = (-2.0*Z3B3+ZY34+ZY33)*B3SQ
          P12 = (2.0*(ZX3B3-ZXY33)+ZX3B3-ZXY34)*B3
          P13 = (-2.0*ZX3B3+ZXY34+ZXY33)*B3SQ
          P20 = (2.0*(Z3A3-ZX33)+Z3A3-ZX43)*A3
          P21 = (2.0*(ZY3A3-ZXY33)+ZY3A3-ZXY43)*A3
          P22 = (3.0*(A+E)+D)*A3*B3
          P23 = (-3.0*E-B-D)*A3*B3SQ
          P30 = (-2.0*Z3A3+ZX43+ZX33)*A3SQ
          P31 = (-2.0*ZY3A3+ZXY43+ZXY33)*A3SQ
          P32 = (-3.0*E-C-D)*B3*A3SQ
          P33 = (D+E+E)*A3SQ*B3SQ
C COMPUTATION OF THE POLYNOMIAL
          DO 370 JY=1,JYMX
            KV = KV0 + JY
            DY = V(KV) - Y3
            Q0 = P00 + DY*(P01+DY*(P02+DY*P03))
            Q1 = P10 + DY*(P11+DY*(P12+DY*P13))
            Q2 = P20 + DY*(P21+DY*(P22+DY*P23))
            Q3 = P30 + DY*(P31+DY*(P32+DY*P33))
            DO 360 JX=1,JXMX
              KU = KU0 + JX
              DX = U(KU) - X3
              W(KU,KV) = Q0 + DX*(Q1+DX*(Q2+DX*Q3))
  360       CONTINUE
```

```
  370       CONTINUE
            KU0 = KU0 + MX0
  380     CONTINUE
          KV0 = KV0 + MY0
  390 CONTINUE
C NORMAL EXIT
      RETURN
C ERROR EXIT
  400 WRITE (IU0,99999)
      GO TO 520
  410 WRITE (IU0,99998)
      GO TO 520
  420 WRITE (IU0,99997)
      GO TO 520
  430 WRITE (IU0,99996)
      GO TO 520
  440 WRITE (IU0,99995)
      GO TO 520
  450 WRITE (IU0,99994)
      GO TO 520
  460 WRITE (IU0,99993)
      GO TO 480
  470 WRITE (IU0,99992)
  480 WRITE (IU0,99991) IX, X(IX)
      GO TO 520
  490 WRITE (IU0,99990)
      GO TO 510
  500 WRITE (IU0,99989)
  510 WRITE (IU0,99988) IY, Y(IY)
  520 WRITE (IU0,99987) LX0, MX0, NU0, LY0, MY0, NV0
      RETURN
C FORMAT STATEMENTS
99999 FORMAT(1X/23H  ***   LX = 1 OR LESS./)
99998 FORMAT(1X/23H  ***   LY = 1 OR LESS./)
99997 FORMAT(1X/23H  ***   MX = 1 OR LESS./)
99996 FORMAT(1X/23H  ***   MY = 1 OR LESS./)
99995 FORMAT(1X/26H  ***   IMPROPER NU VALUE./)
99994 FORMAT(1X/26H  ***   IMPROPER NV VALUE./)
99993 FORMAT(1X/27H  ***   IDENTICAL X VALUES./)
99992 FORMAT(1X/33H  ***   X VALUES OUT OF SEQUENCE./)
99991 FORMAT(7H   IX =, I6, 10X, 7HX(IX) =, E12.3)
99990 FORMAT(1X/27H  ***   IDENTICAL Y VALUES./)
99989 FORMAT(1X/33H  ***   Y VALUES OUT OF SEQUENCE./)
99988 FORMAT(7H   IY =, I6, 10X, 7HY(IY) =, E12.3)
99987 FORMAT(7H   LX =, I6, 10X, 4HMX =, I6, 10X, 4HNU =, I6/
     * 7H   LY =, I6, 10X, 4HMY =, I6, 10X, 4HNV =, I6/6H ERROR,
     * 30H DETECTED IN ROUTINE    SFCFIT)

      END
```

Mason-modellen

blande.m

Program som forenkler kjøringen av `sensf.m` for ulike blandingsforhold mellom Araldite og metallpulver.

```
function [ut,z] = blande(f,phim,valg);
% [ut,z] = blande(f,phim,valg)
% Kjører sensf.m for ulike blandingsforhold
% mellom metall og Araldite.

% ut    - matrise, flyttall, length(phim) x length(f),
%         resultat fra kjoeringer av sensf.m
% z     - flyttall, spesifikk akustisk impedans
% f     - vektor, flyttall, frekvenser
% phim  - vektor, flyttall, masseandel metall i frontlaget
% valg  - streng, 'aluminium' eller 'kobber'

%% %%% Araldite %%%
ce = 2620;
rhoe = 1160;
betae = rhoe*ce^2;

switch valg
case 'aluminium'
  %% %%% Aluminium %%%
  betaa = 7.5e10;
  rhoa = 2700;
case 'kobber'
  %% %%% Kobber %%%
  betaa = 16e10;
  rhoa = 8900;
end

phi = phim*rhoe./(rhoa + phim*(rhoe-rhoa));
betaeff = (1/betae*(1-phi) + 1/betaa*phi).^(-1);
rhoeff = rhoe*(1-phi) + rhoa*phi;
ceff = sqrt(betaeff./rhoeff);

for ii = 1:length(phim)
  ut(ii,:) = sensf(f,rhoeff(ii),ceff(ii));
end
z = rhoeff.*ceff;
```

tykk.m

Program som forenkler kjøringen av `sensf.m` ved ulike tykkelser på koblingslaget. `sensf.m` må modereres noe for å ta imot de rette argumentene.

```
function [ut,norml] = tykk(f,phim,lk,valg);
% [ut,norml] = tykk(f,phim,lk,valg)
% Kjører senst.m for ulike tykkelser paa koblingslaget.

% senst.m er som sensf.m, men mottar argumentene
% f, rhoeff, ceff, og lk.
%
% ut    - matrise, flyttall, length(lk) x length(f),
%         resultat fra kjoeringer av senst.m
% norml - flyttall, frontlagets tykkelse i boelgelengder
% f     - vektor, flyttall, frekvenser
% phim  - flyttall, masseandel metall i frontlaget
% lk    - vektor, flyttall, tykkelser paa frontlaget
% valg  - streng, 'aluminium' eller 'kobber'


%% %%% Araldite %%%
ce = 2620;
rhoe = 1160;
betae = rhoe*ce^2;

switch valg
  case 'aluminium'
    % %%% Aluminium %%%
    betaa = 7.5e10;
    rhoa = 2700;
  case 'kobber'
    %% %%% Kobber %%%
    betaa = 16e10;
```

```
    rhoa = 8900;
end

phi = phim*rhoe./(rhoa + phim*(rhoe-rhoa));
betaeff = (1/betae*(1-phi) + 1/betaa*phi).^(-1);

rhoeff = rhoe*(1-phi) + rhoa*phi;
ceff = sqrt(betaeff./rhoeff);

fs = 1003.33e3; % Resonansfrekvens
norml = (lk./ceff)*fs;

% Kjoerer senst.m, en modifisert versjon
% av sensf.m.
for ii = 1:length(lk)
  ut(ii,:) = senst(f,rhoeff,ceff,lk(ii));
end
%z = rhoeff.*ceff;
```

sensf.m

Dette programmet implementerer Mason-modellen slik den er omskrevet i avsnitt 1.2.2. Det kan brukes til å finne transduserens admittans, kildefølsomhet og overføringsfunksjon H^{Vu} med ulike koblingslag.

```
function [ssens] = sensf(f,rhok,ckinn)
% [ssens] = sensf(f,rhok,ckinn)
% Mason-modell for skive med frontlag mot vann, luft bak skiven.
% ssens -- kildefoelsomhet
% f -- frekvens
% rhok -- massetetthet koblingslag
% ckinn -- lydfart koblingslag

d = 49.955e-3;   % diameter
V = 1;           % inngangsspenning (amplitude)
A = pi*(d/2)^2;  % areal

% --- Piezokeramisk element ---
c33D = 1.5621e11;
eps33S = 5.8140e-9;
e33  = 13.9142;
rhom = 7700;       % tetthet
Qm   = 1090;       % mekanisk Q-faktor
tand = 0.003;      % elektrisk tapsfaktor
l    = 2.0322e-3;  % tykkelse
fs   = 1003.33e3;  % TE serie-resonansfrekvens
%c33D1 = c33D.*(1 + i./Qm);    % korreksjon for mekanisk tap
C0 = eps33S.*A./l;            % parallellkapasitans
C0 = C0.*sqrt(1 - i.*tand);   % dielektrisk tap
cm = sqrt(c33D1./rhom);       % lydfart
phi = e33.*A./l;              % koblingsfaktor

% --- Frontlag ---
Qmk = 200;
%lk = ckinn./fs*1/4;          % tykkelse
lk = 5.38e-4;
%ck = ckinn*sqrt(1 + i/Qmk);  % korreksjon for mekanisk tap
ck = ckinn;

% --- Vann ---
c0   = 1500;
rho0 = 1000;


w = 2.*pi.*f;
km = w./cm;
kk = w./ck;
Z0m = rhom.*cm.*A;
ZA = (i.*Z0m./phi.^2).*tan(km.*l./2);
ZB = Z0m./(i.*phi.^2.*sin(km.*l));
Z0k = rhok.*ck.*A;
ZAK = (i.*Z0k./phi.^2).*tan(kk.*lk./2);
ZBK = Z0k./(i.*phi.^2.*sin(kk.*lk));
ZR = A.*rho0.*c0./phi^2;

ZC = i./(w.*C0);
%YT = i.*w.*C0 + 1./(ZC + ZB + ZA./2); % vakuum

ZAKR = ZAK + ZR;
ZL = ZAK + (1./(1./ZAKR + 1./ZBK));
%ZL1 = ZL.*ZBK./(ZL-ZAK);
ZL2 = ZL.*ZAKR./(ZL-ZAK);
```

```
ZAL1 = ZA;
ZAL2 = ZA + ZL;
ZBC = ZB + ZC;
ZP = 1./(1./ZAL2 + 1./ZAL1) + ZBC;

YT = i.*w.*C0 + 1./ZP; % med last

%%ZM1 = ZAL1.*ZP./(ZP-ZBC);
ZM2 = ZAL2.*ZP./(ZP-ZBC);

u2 = V./phi./ZM2;
ur = u2.*ZL./ZL2;
p = rho0.*c0.*ur;
ssens = p./V;
```

Behandling av måledata

naal.m

Kalibreringsprogram for Precision Acoustics-nålhydrofon.

```
function [trykk] = naal(spenning,frekvens)
% trykk = hydrofon(spenning,frekvens)
%
% Regner om signalet fra Precision Acoustics-naalhydrofonen ved hjelp av den oppgitte
% følsomheten.
%
% spenning - vektor [V]
% frekvens - flyttall [kHz]
%
% trykk - vektor [Pa]

S = [.6478 .563 .5024 .4755 .3473 .4257 .4144 .3893 .3939 .4059 .3832 .3733 ...
     .3712 .359 .3617 .3621 .3586 .3465 .3432 .3378]*1e-6; %PA [V/Pa]
%usikkerhet_rel = [.14 .14 .14 .14 .14 .15 .14 .14 .15 .16 .15 .15 .17 .17 ...
                   .17 .17 .17 .17 .17 .17];
%usikkerhet = S.*usikkerhet_rel;
f = [1:20]*1e3;

sens = spline(f,S,frekvens);

trykk = spenning/sens;
```

preamp4.m

Kalibreringsprogram for Panametrics-forforsterkeren.

```
function kalibrert = preamp4(ukalibrert)
% kalibrert = preamp4(ukalibrert)
%
% Regner om spenninger fra forforstekerens utgang til inngangen
% ved 40 dB forsterkning. 50 Ohms last på begge terminalene.
% LeCroy måledata fra preamp.mat brukes.
% Programmet bruker 3. ordens polynomtilpasning ved hjelp av minste kvadraters metode.
% preamp.mat gjelder for 1005 kHz.

load preamp.mat
P = polyfit(ut,inn,3);
kalibrert = polyval(P,ukalibrert);
```

hydrofon.m

Kalibreringsprogram for GEC-Marconi-hydrofonen.

```
function [trykk] = hydrofon(spenning,frekvens,kabel)
% trykk = hydrofon(spenning,frekvens,kabel)
%
% Regner om signalet fra GEC-Marconi-hydrofonen ved hjelp av den oppgitte
% frittfelts følsomheten. Fra kabelen er bare den kapasitive lasten tatt med.
%
% spenning - vektor [V]
% frekvens - flyttal [kHz]
```

```
% kabel    - flyttall [m] (Lengden på kabelen mellom hydrofonen og oscilloskopet (1MOhm))
%
% trykk - vektor [Pa]

S = [.035 .035 .036 .037 .037 .038 .039 .040 .041 .043 .044 .046 .047 .050 .052 ...
     .055 .057 .059 .061 .061]*1e-6; % GEC-Marconi
f = [1:20]*1e3;
R = [160 93 65 51 40 35 31 28 26 23 22 21 20 20 18 18 17 17 16 15];
X = -[1800 930 640 490 390 330 280 250 220 200 180 170 150 140 130 120 110 110 98 92];

if kabel > 0
  c = 100e-12*kabel + 15e-12;
else
  c = 320e-12;
end

r = spline(f,R,frekvens);
x = spline(f,X,frekvens);
sens0 = polyval(polyfit(f,S,2),frekvens);

ch = -1/(2*pi*frekvens*1e3*x);
sens = sens0*ch/(ch + c);

trykk = spenning/sens;
```

glattut.m

Erstatter defekte målepunkt med middelverdien av nabopunktene.

```
function inn = glattut(inn)
% ut = glattut(inn)
%
% Fjerner glipper i målinger fra LeCroy-oscilloskopet og
% erstatter dem med middelverdien av nabopunktene.

if size(inn,1) == 1 | size(inn,2) == 1
   for ii = 2:length(inn)-1
      if abs(inn(ii)-0.5*(inn(ii-1)+inn(ii+1))) > 0.08*0.5*(inn(ii-1)+inn(ii+1))
         inn(ii) = 0.5*(inn(ii-1) + inn(ii+1));
      end
   end
else
   for jj = 1:size(inn,1)
      for ii = 2:size(inn,2)-1
         if abs(inn(jj,ii)-0.5*(inn(jj,ii-1)+inn(jj,ii+1))) > 0.06*0.5*(inn(jj,ii-1)+inn(jj,ii+1))
            inn(jj,ii) = 0.5*(inn(jj,ii-1) + inn(jj,ii+1));
         end
      end
   end
end
```

trans.m

Program for Fouriertransformasjon av en matrise med en målt bølgeform i hver rad. Kaller et av kalibreringsprogrammene `naal.m` og `hydrofon.m` for å kompensere for hydrofonens følsomhet.

```
% trans.m
%
% Fouriertransformasjon og kalibrering
% av et maalt signal.


addpath /heim/audun/maalinger/kalib/
addpath /heim/audun/maalinger/styring/
% load AKSE.MAT

filnavn = 'a380h';

m = 455;
n = 100;

figure;
subplot(2,1,1);
plot([Uax(n,300:m) Uax(n,1:100)]);

subplot(2,1,2);
```

```
for ii = 1:size(Uax,1)
  tr(ii,:) = fft([Uax(ii,1:m) Uax(ii,1:m)]);
end
N = size(tr,2);
plot(abs(tr(n,1:120)));

streng = ['save ' filnavn ' z vdiv amp0 '];
amp0 = tr(:,1)/N;
for ii = 1:20
  eval(['amp' num2str(ii) ' = naal(2/N*abs(tr(:,1+10*ii)),1100*ii);']);
%  eval(['amp' num2str(ii) ' = hydrofon(2/N*abs(tr(:,1+10*ii)),1100*ii,1);']);
  eval(['fase' num2str(ii) ' = -atan(real(tr(:,1+10*ii))./imag(tr(:,1+10*ii)));']);
  if ii < 7
    eval(['amp' num2str(ii) ' = glattut(amp' num2str(ii) ');']);
    eval(['fase' num2str(ii) ' = glattut(fase' num2str(ii) ');']);
  end
end

figure;
hold on;
for ii = 1:10
  eval(['plot(amp' num2str(ii) ');']);
  streng = [streng 'amp' num2str(ii) ' fase' num2str(ii) ' '];
end
hold off;

% Lagrer aksharm
eval(streng);
```

Bibliografi

[1] L. E. Kinsler, A. R. Frey, A. B. Coppens og J. V. Sanders. *Fundamentals of Acoustics*. Wiley, 3. utg. utgave, 1982.

[2] Sonic boom. USAF fact sheet 96-03.

[3] Omtale av sober-programmet. www.cordis.lu/growth.

[4] M. Vestrheim. Akustiske transdusere. Forelesningsnotater og laboratorieoppgaver for emnet FYS 272, Fysisk institutt, UiB.

[5] IEEE std. 176-1987 — standard on piezoelectricity, 1987.

[6] C. S. DeSilets, J. D. Fraser og G. S. Kino. The design of efficient broad-band piezoelectric transducers. *IEEE trans. son. ult.*, SU-25(3):115–125, 1978.

[7] H. Hobæk. Ikkelineære vekselvirkninger mellom to lydstråler: En eksperimentell undersøkelse. Hovedfagsoppgave, Fysisk institutt, UiB, 1967.

[8] N. G. Pace og P. D. Thorne. Efficiency of face-plated underwater acoustic transducers. *Ultrasonics*, 21:65–69, 1983.

[9] O. Krauß, R. Gerlach og J. Fricke. Experimental and theoretical investigations of SiO_2-aerogel matched piezo-transducers. *Ultrasonics*, 32(3):217–222, 1994.

[10] J. H. Goll og B. A. Auld. Multilayer impedance matching schemes for broadbanding of water loaded piezoelectric transducers and high q electric resonators. *IEEE trans. son. ult.*, SU-22(1):52–53, 1975.

[11] J. H. Goll. The design of broad-band fluid-loaded ultrasonic transducers. *IEEE trans. son. ult.*, SU-26(6):385–393, 1979.

[12] S. J. H. van Kervel og J. M. Thijssen. A calculation scheme for the optimum design of ultrasonic transducers. *Ultrasonics*, 21:134–140, 1983.

[13] N. Lamberti, G. Caliano, A. Iula og M. Pappalardo. A new approach for the design of ultrasono-therapy transducers. *IEEE trans. son. ult.*, 44(1):77–84, 1997.

[14] G. Kossoff. The effects of backing and matching on the performance of piezoelectric ceramic transducers. *IEEE transactions on sonics and ultrasonics*, SU-13(1):20–30, 1966.

[15] M. Vestrheim. Akustiske målesystemer. Forelesningsnotater for emnet FYS 373, Fysisk institutt, UiB.

[16] J. Kocbach. *Finite Element Modelling of Ultrasonic Piezoelectric Transducers*. Doktorgradsoppgave, Fysisk institutt, UiB, 2000.

[17] Ø. Nesse. *Sound Propagation in Emulsions*. Doktorgradsoppgave, Fysisk institutt, UiB, 1998.

[18] M. F. Hamilton og D. T. Blackstock, redaktører. *Nonlinear Acoustics*. Academic Press, 1997.

[19] H. Hobæk. Nonlinear acoustics. Forelesningsnotater for emnet FYS 372, Fysisk institutt, UiB.

[20] J. Berntsen. Numerical calculations of finite amplitude sound beams. I M. F. Hamilton og D. T. Blackstock, redaktør, *Frontiers of Nonlinear Acoustics*, side 191–196. Elsevier, London, 1990.

[21] J. Naze Tjøtta, S. Tjøtta og E. H. Vefring. Effects of focusing on the nonlinear interaction between two collinear finite amplitude sound beams. *J. Acoust. Soc. Am.*, 89:1017–1027, 1991.

[22] S. Nachef, D. Cathignol, A. M. Berg og S. Tjøtta. Investigation of a high intensity sound beam from a plane transducer. experimental and theoretical results. *J. Acoust. Soc. Am.*, 98(4):2303–2323, 1995.

[23] B. Ystad og J. Berntsen. Numerical solution of the KZK equation for focusing sources. *Acta acustica*, 3:323–330, 1995.

[24] J. Naze Tjøtta og S. Tjøtta. Model equation and boundary conditions for the sound field from a high frequency, strongly curved and highly intense transducer. *Acta acustica*, 1:69–87, 1993.

[25] B. Ystad og J. Berntsen. Numerical solution of parabolic equations for strongly curved focusing sources. *Acustica/Acta acustica*, 82:698–706, 1996.

[26] J. Naze Tjøtta, S. Tjøtta og E. H. Vefring. Propagation and interaction of two collinear finite amplitude sound beams. *J. Acoust. Soc. Am.*, 88:2859–2870, 1990.

[27] M. F. Hamilton, J. N. Tjøtta og S. Tjøtta. Nonlinear effects in the farfield of a directive sound source. *J. Acoust. Soc. Am.*, 78(1):202–214, 1985.

[28] C. F. Gerald og P. O. Wheatley. *Applied Numerical Analysis*. Addison-Wesley, 5 utgave, 1994.

[29] J. Berntsen og E. H. Vefring. Numerical computation of a finite amplitude sound beam. Rapport 81, Matematisk institutt, UiB, 1986.

[30] S. I. Aanonsen, T. Barkve, J. Naze Tjøtta og S. Tjøtta. Distortion and harmonic generation in the nearfield of a finite amplitude sound beam. *J. Acoust. Soc. Am.*, 75:749–768, 1984.

[31] R. Fardal. Endelig element analyse av elektriske egenskaper til piezoelektriske skiver. Hovedfagsoppgave, Fysisk institutt, UiB, 2002.

[32] F. Atterås. Materialkonstantar for piezokeramiske elementer. Hovedfagsoppgave, Fysisk institutt, UiB, 1998.

[33] Guide to dynamic measurements of piezoelectric ceramics with high electromechanical coupling, 1976. IEC standard, publication 483.

[34] Ferroperm material constants. Publisert på www.ferroperm-piezo.com, 2000.

[35] J. Kocbach. Endelig element-modellering av piezoelektriske skiver. Hovedfagsoppgave, Fysisk institutt, UiB, 1996.

[36] L. Pedersen. Acoustic properties. Publisert på www.ultrasonic.com.

[37] R. Kippersund. Eksperimentell undersøkelse av ikkelineær lydforplantning fra plane og fokuserende kilder. Hovedfagsoppgave, Fysisk institutt, UiB, 2002.

[38] GEC-Marconi. PVDF membrane hydrophone information sheet. Kalibreringsdata fra National Physical Laboratory 29. mars 1995.

[39] Precision Acoustics Ltd. Hydrophone calibration certificate. Kalibreringsdata for nålhydrofon 17. desember 2001.

[40] C. S. Williams. *Designing Digital Filters*. Prentice-Hall, 1985.

[41] A. Papoulis. *Signal Analysis*. McGraw-Hill, 1984.

[42] The MathWorks, Inc. Matlab 6 hjelpefiler, 2000.

[43] B. G. Lucas og T. G. Muir. The field of a focusing source. *J. Acoust. Soc. Am.*, 72(4):1289–1296, 1982.