

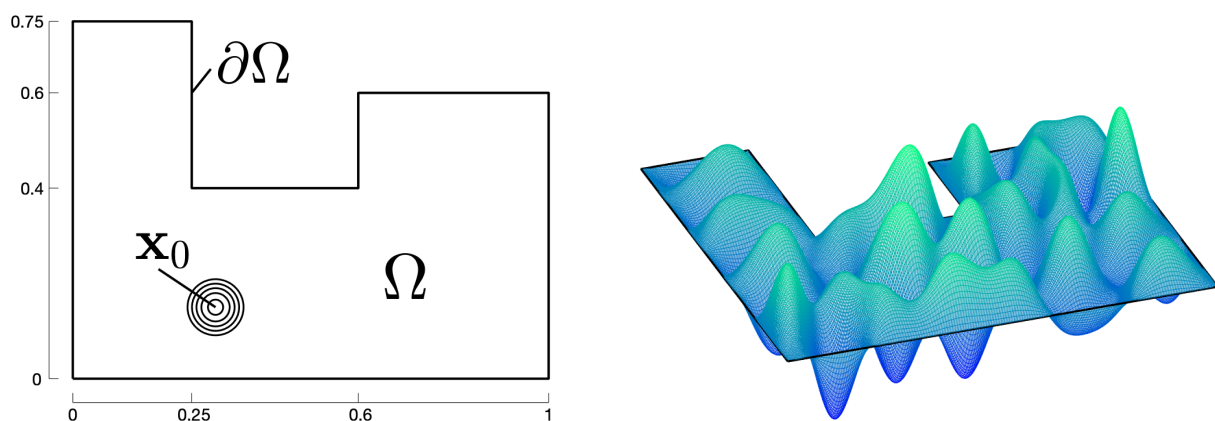


KTH Engineering Sciences

Projekt Hitta ljudkällor

Problem

Vi vill hitta positionen för en ljudkälla i ett slutet rum, givet information om ljudet som hörs vid rummets väggar. (Man kan tänka sig det sitter mikrofoner längs väggarna där man kan lyssna på ljudet.) Detta är ett klassiskt problem inom akustiken. Om man antar att ljudet genereras av en källa av känd typ kan positionen beräknas; annars går det i allmänhet inte. Notera att ljudet reflekteras vid väggarna. Ljudvågorna studsar därför runt i rummet och bildar ett mycket komplicerat mönster (se högra bilden nedan).



Figur 1. Vänster: Skiss av rummet. Höger: Exempel på lösning $u(\mathbf{x})$, när $\mathbf{x}_0 = (0.3, 0.15)$ och $\omega = 40$.

Matematisk bakgrund

Vi låter $\Omega \subset \mathbb{R}^2$ vara ett öppet begränsat område som beskriver rummet. Dess dimensioner ges i bilden ovan. Väggarna utgör randen till Ω , som vi kallar $\partial\Omega$. Fysikaliskt är ljud små tryckvariationer, som ändras sig snabbt i tid och rum. Vi låter $p(t, \mathbf{x})$ vara avvikelser från bakgrundstrycket vid tiden t och positionen $\mathbf{x} = (x, y)$. Om vi antar att ljudet har en fix frekvens f kan vi då skriva

$$p(t, \mathbf{x}) = u(\mathbf{x}) \cos(\omega t), \quad \omega = 2\pi f,$$

där u beskriver ljudets amplitud vid \mathbf{x} . Vi inför också källfunktionen $S = S(\mathbf{x})$ som visar hur mycket ljud som genereras vid \mathbf{x} . För en punktkälla är tex S noll överallt förutom i ett litet område runt en punkt.

Man kan visa att amplituden u kommer att uppfylla en *partiell differentialekvation*, kallad Helmholtz ekvation, som lyder

$$\Delta u(\mathbf{x}) + \omega^2 u(\mathbf{x}) = S(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega. \quad (1)$$

1 (6)

Vid rummets väggar kan vi anta att $u = 0$. Detta ger *randvillkoret*,

$$u(\mathbf{x}) = 0, \quad \forall \mathbf{x} \in \partial\Omega. \quad (2)$$

Tillsammans utgör (1) och (2) ett väldefinierat problem (för nästan alla ω) med en entydig lösning u .

I problemet vi vill lösa antar vi att $S(\mathbf{x}) = aS_0(\mathbf{x} - \mathbf{x}_0)$ där funktionen S_0 är känd (och bara nollskild nära origo) men källstyrkan a och källans position \mathbf{x}_0 är okända. Vidare antar vi att våra mikrofoner kan registrera amplitudens normalderivata vid väggarna, där $u = 0$. Mer precist, antar vi att vi känner

$$g(\mathbf{x}) = \frac{\partial u(\mathbf{x})}{\partial n} := \nabla u(\mathbf{x}) \cdot \hat{n}(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega,$$

där $\hat{n}(\mathbf{x})$ är den utåtriktade enhetsnormalen till Ω vid \mathbf{x} . Från kunskapen om S_0 , g och Helmholtz ekvation vill vi hitta källans styrka a och position \mathbf{x}_0 .

Att lösa Helmholtz ekvation (1) och (2) är normalt ganska komplicerat och kräver att man använder numeriska metoder. Här är vi intresserade av något som kan ses som det omvända: Vi får givet (en del av) lösningen till Helmholtz, och vill bestämma funktionen S i differentialekvationen. Detta brukar kallas för ett *inverst problem*, och problemet i det här projektet kallas det inversa källproblemet för Helmholtz. För att lösa inversa problem måste man vanligtvis lösa Helmholtz ekvation direkt, med kända data, många gånger. För det här källproblemet kan man dock göra det enbart genom integrera kända funktioner och minstakvadratanpassa resultaten till en icke-linjär modell.

Del 1 – Bekanta er med problemet

Indata till problemet som ni ska lösa är funktionen g , amplitudens normalderivata vid väggarna. Istället för att mäta upp amplituden fysiskt använder vi i projektet en simulering, där Helmholtz ekvation löses numeriskt och g avläses från resultatet.

Programmet som gör simuleringen heter `hhsolver.m` och ni hittar det i Canvas. Utöver g ger programmet även som sidosresultat hela lösningen u till Helmholtz i rummet. Denna lösning används egentligen aldrig i projektet, men den kan ge en förståelse och intuition för problemet ni ska lösa. Börja därför med att provköra `hhsolver.m` och titta på lösningarna för olika fall.

Syntaxen för `hhsolver.m` är

`[Bound,Sol] = hhsolver(omega,S,N),`

där

- Indata är:

- `omega` = vinkelfrekvensen ω ,
- `S` = en handle till källfunktionen S ; funktionen ska ta x och y som argument.
- `N` = antal approximationspunkter i x -led; $\Delta x = 1/N$ är upplösningen på den beräknade lösningen. Punkterna är ekvidistant fördelade så att $\Delta y = \Delta x$. (Ju större N desto mer beräkningskrävande är simuleringen.)

- Utdata är:

- `Sol` = en Matlab-**struct** som innehåller lösningen i Ω . `Sol.x` och `Sol.y` är koordinaterna för punkterna i rummet. `Sol.u` är lösningen u evaluerad i punkterna.
- `Bound` = en Matlab-**struct** som innehåller lösningens normalderivata på randen $\partial\Omega$. `Bound.x` och `Bound.y` är koordinaterna för randpunkterna. `Bound.un` är lösningens normalderivata $\partial u / \partial n = g$ evaluerad i randpunkterna. `Bound.s` är en båglängdsparameter som är noll i origo och sedan löper motsols runt rummet. Den används när man beräknar kurvintegraler längs randen.

I Canvas finns filen `projexempel.m` som visar exempel på hur man beräknar en lösning med `hhsolver.m` och sedan visualiserar den med plottar av olika typer. Prova att variera källfunktionen S_0 , källans position \mathbf{x}_0 , upplösningen N och frekvensen ω . Notera att för högre frekvens ω , och smalare punktkälla S_0 kommer man behöva ta större N för att få noggranna resultat. Som utgångspunkt kan ni tex välja $a \approx 1$, $\omega \approx 20$, $N \approx 200$ och \mathbf{x}_0 någonstans inuti rummet. Källfunktionen kan vara

$$S_0(\mathbf{x}) = e^{-w|\mathbf{x}|^2}, \quad w = 400.$$

Del 2 – Teoretiska uppgifter

1. Antag att funktionen v satisfierar Helmholtz ekvation (1) med högerledet noll, dvs

$$\Delta v(\mathbf{x}) + \omega^2 v(\mathbf{x}) = 0, \quad \forall \mathbf{x} \in \Omega, \quad (3)$$

men inte nödvändigtvis randvillkoret (2). Visa att

$$\int_{\partial\Omega} v(\mathbf{x})g(\mathbf{x})ds = \int_{\Omega} S(\mathbf{x})v(\mathbf{x})d\mathbf{x}.$$

Hint: Multiplicera först (1) med v och integrera över Ω . Det ger

$$\int_{\Omega} v(\mathbf{x})[\Delta v(\mathbf{x}) + \omega^2 v(\mathbf{x})]d\mathbf{x} = \int_{\Omega} v(\mathbf{x})S(\mathbf{x})d\mathbf{x}.$$

Använd sedan Greens sats och utnyttja (3) samt randvillkoret (2).

2. Visa att funktionerna

$$v_c(\mathbf{x}; \alpha) = \cos(\omega(x \cos \alpha + y \sin \alpha)), \quad v_s(\mathbf{x}; \alpha) = \sin(\omega(x \cos \alpha + y \sin \alpha)),$$

båda satisfierar (1) med $S \equiv 0$, för alla α . Funktionerna motsvarar plana ljudvågor som utbreder sig i riktningen α . Notera: Med komplex notation kan vi också skriva

$$e^{i\omega \hat{s} \cdot \mathbf{x}} = v_c + i v_s, \quad \hat{s}(\alpha) = \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix}.$$

3. Antag att källan S ligger en bit från väggarna så att $S(\mathbf{x}) = 0$ för alla \mathbf{x} utanför Ω .¹ Visa att vi då har sambanden

$$\begin{aligned} I_c(\alpha) &= aF_c(\alpha)v_c(\mathbf{x}_0; \alpha) - aF_s(\alpha)v_s(\mathbf{x}_0; \alpha), \\ I_s(\alpha) &= aF_s(\alpha)v_c(\mathbf{x}_0; \alpha) + aF_c(\alpha)v_s(\mathbf{x}_0; \alpha), \end{aligned}$$

där

$$I_c(\alpha) := \int_{\partial\Omega} v_c(\mathbf{x}, \alpha)g(\mathbf{x})ds, \quad I_s(\alpha) := \int_{\partial\Omega} v_s(\mathbf{x}, \alpha)g(\mathbf{x})ds,$$

och

$$F_c(\alpha) := \int_{\mathbb{R}^2} v_c(\mathbf{x}, \alpha)S_0(\mathbf{x})d\mathbf{x}, \quad F_s(\alpha) := \int_{\mathbb{R}^2} v_s(\mathbf{x}, \alpha)S_0(\mathbf{x})d\mathbf{x}.$$

Hint: Använd resultaten i uppgifterna 1 och 2 ovan. Utnyttja informationen om S . Variabelbytet $\mathbf{y} = \mathbf{x} - \mathbf{x}_0$ är användbart, och även additionsformlerna för trigonometriska funktioner.

4. En *isotrop* källa genererar lika mycket ljud i alla riktningar. Det betyder att S_0 bara beror på avståndet $|\mathbf{x}|$ till origo, dvs $S_0 = S_0(|\mathbf{x}|)$. Visa att $F_c(\alpha) \equiv \text{konstant}$ för isotropa källor. Visa också att $F_s(\alpha) \equiv 0$ när $S_0(-\mathbf{x}) = S_0(\mathbf{x})$.

¹ Matematiskt skriver man $\text{supp}(S) \subset \Omega$, där " $\text{supp}(S)$ " betyder funktionens *stöd* (support), definierat som förslutningen av mängden punkter där S är nollskild, $\text{supp}(S) := \{\mathbf{x} \in \mathbb{R}^2 \mid S(\mathbf{x}) \neq 0\}$.

Del 3 – Inversa problemet med isotrop källa

Ni ska nu hitta källans styrka och position numeriskt, när källan är isotrop och S_0 ges av

$$S_0(\mathbf{x}) = \cos(20|\mathbf{x}|) e^{-1000|\mathbf{x}|^2}.$$

Från analysen ovan vet vi då att

$$I_c(\alpha) = aF_c v_c(\mathbf{x}_0; \alpha) = aF_c \cos(\omega(x_0 \cos \alpha + y_0 \sin \alpha)), \quad (4)$$

för alla α . Här beror F_c alltså på S_0 men inte på α .

1. Beräkna konstanten F_c med numerisk integration i två dimensioner. Verifiera (numeriskt) att den inte beror på α . (Använd $\omega = 19$ här.)
2. Välj själv styrka och position för källan och beräkna g med `hhsolver.m`. Använd $\omega = 19$ och N tillräckligt stort (minst 100). Källan ska ligga en bit in i rummet så att $S \approx 0$ vid väggarna.
3. Från ditt g , beräkna $I_c(\alpha)$ med hjälp av numerisk integration av randintegralen som definierar I_c . (Båglängdsvärdena `Bound.s` är användbara här.) Beräkna $I_c(\alpha)$ för M värden $\alpha = \alpha_1, \dots, \alpha_M$, där M är minst 10.
4. Använd Gauss-Newtons metod för att anpassa dessa värden till en cosinus-planvåg, dvs hitta konstanterna \tilde{a} , \tilde{x}_0 , \tilde{y}_0 så att

$$I_c(\alpha_j) \approx \tilde{a} \cos(\tilde{x}_0 \cos \alpha_j + \tilde{y}_0 \sin \alpha_j), \quad j = 1, \dots, M.$$

Källans styrka och position ges sedan av $a \approx \tilde{a}/F_c$, $x_0 \approx \tilde{x}_0/\omega$ och $y_0 \approx \tilde{y}_0/\omega$.

Använd startgissningar till Gauss-Newton som motsvarar värden nära dina valda a , x_0 och y_0 .

Hint: För att verifiera att allt blivit rätt med F_c och $I_c(\alpha)$ kan ni plotta vänsterledet $I_c(\alpha)$ och högerledet $aF_c v_c(\mathbf{x}_0; \alpha)$ i (4) som funktion av α , med de kända värdena på x_0 , y_0 och a . Det bör ge två identiska kurvor.

5. Antag att ni inte känner källans form och frekvensen exakt. För frekvensen vet ni bara att $\omega = 19.0 \pm 0.2$ och för källan att

$$S_0(\mathbf{x}) = \cos(\rho|\mathbf{x}|) e^{-1000|\mathbf{x}|^2}.$$

där parametern $\rho = 20 \pm 1$. Använd experimentell störningsräkning för att räkna fram den resulterande osäkerheten i x_0 , y_0 och a . (Obs! Störningen av S_0 och ω ska bara göras i lösningsalgoritmen. Värdena på indata g som beräknats med `hhsolver.m` ska vara oförändrade.)

Vi antar fortsättningsvis att S_0 är känd utan osäkerhet.

6. Lägg på brus (normalfördelade slumpstal) på den uppmätta normalderivatan av amplituden g och undersök hur algoritmen klarar av detta. Ni kan tex använda Matlab-kommandona

```
>> gnoise=g+max(abs(g))*randn(size(g))*noiselevel;
```

Prova med olika brusnivåer `noiselevel`, från tex 10^{-2} till 1 eller tom större. Hur stort kan bruset vara utan att lösningen blir helt fel? (Plotta gärna `g` och `gnoise` som funktion av s för att visualisera brusnivån.)

7. Det kan vara svårt att hitta bra startgissningar till Gauss-Newton, speciellt när frekvensen är hög. För att få en startgissning kan man utnyttja sambanden

$$\frac{\partial v_c(\mathbf{x}_0, \alpha)}{\partial \alpha} = \omega(x_0 \sin \alpha - y_0 \cos \alpha) v_s(\mathbf{x}_0, \alpha) \Rightarrow I'_c(\alpha) = \omega(x_0 \sin \alpha - y_0 \cos \alpha) I_s(\alpha).$$

Genom att sätta in $\alpha = 0$ och $\alpha = \pi/2$ ger det, när $x_0 \neq 0$, $y_0 \neq 0$ och g brusfri, att

$$x_0 = \frac{I'_c(\pi/2)}{\omega I_s(\pi/2)}, \quad y_0 = -\frac{I'_c(0)}{\omega I_s(0)}, \quad a = \frac{I_c(\pi/2)}{F_c v_c(\mathbf{x}_0; \pi/2)} = \frac{I_c(\pi/2)}{F_c \cos(\omega y_0)}.$$

Beräkna x_0 , y_0 och a på detta sätt. Använd numerisk integration för I_c och I_s som tidigare, och numerisk derivering för I'_c .

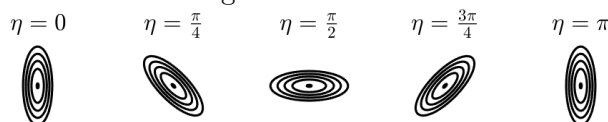
Testa dessa startgissningar med Gauss-Newton för problemen i 4 och 6 ovan. Hur mycket brus klarar algoritmen av nu (när man inte antar att källans läge är känt)?

Startgissningarna kan också beräknas från andra värden på α . Fler än två α -värden kan användas; (x_0, y_0) får då minstakvadratanpassas. När g innehåller brus kan det ge robustare värden. Experimentera gärna med denna metod. (Frivilligt!)

8. På hemsidan finns ett antal filer med okända isotropa källor av olika svårighetsgrad. Testa att lösa dessa. Ni behöver inte klara av alla!
9. I den sista uppgiften byter ni roll och är nu de som genererar ljudet. Ni vill försvåra för de som ska lokalisera källan och byter därför, i hemlighet, ut den mot en *anisotrop* källa, som inte skapar lika mycket ljud i alla riktningar, nämligen

$$S_1(\mathbf{x}, \eta) = \cos(15|\mathbf{x}|) e^{-200[|\mathbf{x}|^2 + 10(\mathbf{x}^T \hat{s}(\eta))^2]}, \quad \hat{s}(\eta) = \begin{pmatrix} \cos \eta \\ \sin \eta \end{pmatrix},$$

där η bestämmer källans orientering. Källan är skissad i bilden nedan för olika η .



Era avlyssnare tror fortfarande att källan är S_0 och applicerar algoritmen från de tidigare uppgifterna, men indatan g beräknad med `hhsolver.m` är nu genererad av S_1 istället för S_0 , och beror på orienteringen η . De kommer då få en lite felaktig position, som också beror på η . Vi kallar den felaktiga positionen $\tilde{\mathbf{x}}_0(\eta) \neq \mathbf{x}_0$. Ni vill hitta orienteringen η som maximerar avvikelsen $E(\eta) = |\tilde{\mathbf{x}}_0(\eta) - \mathbf{x}_0|$ för att förvirra era avlyssnare så mycket som möjligt. (Felet i a ignoreras.)

Använd Gyllene snittet-sökning för att hitta det $\eta = \eta^*$ som maximerar $E(\eta)$ när $\mathbf{x}_0 = (0.80, 0.35)$, $a = 1$ och $\omega = 29$.

Hint: För varje evaluering av $E(\eta)$ måste ni generera nya indata g med `hhsolver.m` och sedan köra er tidigare kod för att få fram $\tilde{\mathbf{x}}_0(\eta)$. Maximum ligger i intervallet $[\pi/2, \pi]$ och $E(\eta)$ är unimodal i detta intervall. Ni behöver ta $N = 400$ eller mer för att få tillräcklig noggrannhet.