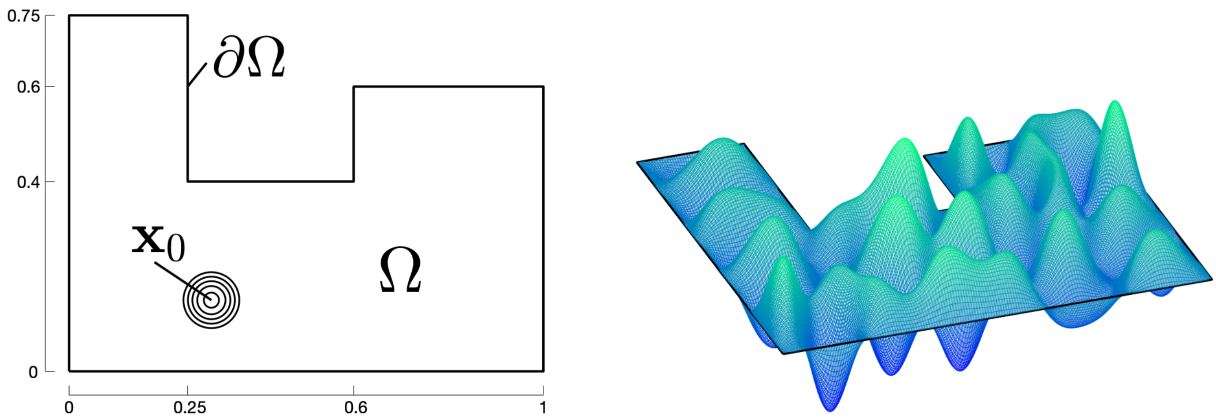
KTH Engineering Sciences

Projekt

Ljudvågor i ett slutet rum

I det här projektet ska ni studera ljudvågor som genereras av en källa i ett slutet rum. Ljudet reflekteras vid väggarna varför vågorna studsar runt i rummet och kan bilda ett mycket komplicerat mönster (se högra bilden nedan). Huvudproblemet handlar om att hitta positionen för ljudkällan givet information om ljudet som hörs vid rummets väggar. (Man kan tänka sig att det sitter mikrofoner längs väggarna där man kan lyssna på ljudet.) En annan uppgift är att placera ljudkällan på ett sätt som gör en del av rummet så tyst som möjligt.



Figur 1. Vänster: Skiss av rummet. Höger: Exempel på lösning $u(\boldsymbol{x})$, när ljudkällan är vid $\boldsymbol{x}_0 = (0.3,\, 0.15)$ 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 ändrar sig snabbt i tid och rum. Vi låter $p(t, \boldsymbol{x})$ vara avvikelsen från bakgrundstrycket vid tiden t och positionen $\boldsymbol{x} = (x, y)$. Om vi antar att ljudet har en fix frekvens f kan vi skriva

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

där u beskriver ljudets amplitud vid $\boldsymbol{x}$. Vi inför också källfunktionen $S = S(\boldsymbol{x})$ som visar hur mycket ljud som genereras vid $\boldsymbol{x}$. 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

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

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

$$u(\boldsymbol{x}) = 0, \qquad \forall \boldsymbol{x} \in \partial\Omega. \tag{2}$$

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

Problem

Er primära uppgift är att hitta ljudkällans läge givet information från mikrofoner längs rummets väggar. 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. Vi antar därför att $S(\boldsymbol{x}) = aS_0(\boldsymbol{x} - \boldsymbol{x}_0)$ där funktionen S_0 är känd (och bara nollskild nära origo) men källstyrkan a och källans position $\boldsymbol{x}_0$ är okända. Vidare antar vi att våra mikrofoner kan registrera amplitudens normalderivata vid väggarna, där $u = 0$. Mer precist, antar vi att vi känner

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

där $\hat{n}(\boldsymbol{x})$ är den utåtriktade enhetsnormalen till Ω vid $\boldsymbol{x}$. Från kunskapen om S_0, g och Helmholtz ekvation, inklusive ω, vill vi hitta källans styrka a och position $\boldsymbol{x}_0$. För att förenkla problemet antar vi slutligen att källan är *isotrop*, vilket betyder att den genererar lika mycket ljud i alla riktningar, dvs S_0 beror bara på avståndet från origo och vi kan skriva $S_0 = S_0(|\boldsymbol{x}|)$.

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 det inversa 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. Simuleringen används också när ni ska hitta en optimal placering för källan som minimerar ljudnivån i en del av rummet.

Programmet som gör simuleringen heter `hhsolver.m` och ni hittar det i Canvas. Utöver g ger programmet även som sidoresultat hela lösningen u till Helmholtz i rummet. (Denna lösning används bara i den andra delen av projektet.) Börja med att provköra `hhsolver.m` och titta på lösningarna i olika fall. Målet är att få förståelse och intuition för hur ljudvågorna ser ut.

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 $\boldsymbol{x}_0$, 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 $\boldsymbol{x}_0$ någonstans inuti rummet. Källfunktionen kan vara

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

Del 2 – Teoretiska uppgifter

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

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

men inte nödvändigtvis randvillkoret (2). Visa att

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

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

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

Använd sedan partiell integration (i 2D, två gånger), utnyttja (3) samt randvillkoret (2).

2. Visa att funktionen

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

satisfierar (3), för alla α. Samma sak gäller

$$v_s(\boldsymbol{x};\alpha) = \sin(\omega(x\cos\alpha + y\sin\alpha)).$$

Funktionerna motsvarar plana ljudvågor som utbreder sig i riktningen α. Notera att man med komplex notation kan skriva

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

3. För alla α gäller att

$$\int_{\mathbb{R}^2} S_0(|\boldsymbol{x}|)v_s(\boldsymbol{x},\alpha)d\boldsymbol{x} = 0, \qquad \int_{\mathbb{R}^2} S_0(|\boldsymbol{x}|)v_c(\boldsymbol{x},\alpha)d\boldsymbol{x} = \int_{\mathbb{R}^2} S_0(|\boldsymbol{x}|)\cos(\omega x)d\boldsymbol{x} =: \eta.$$

Visa det andra sambandet, om v_c och η! Notera speciellt att den andra integralen, och η, är oberoende av α.

Ledning: Använd polära koordinater.

4. Antag att källan S ligger en bit från väggarna så att $S(\boldsymbol{x}) = 0$ för alla $\boldsymbol{x}$ utanför Ω.[1] Definiera randintegralerna

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

Då gäller följande samband

$$I_c(\alpha) = a\eta v_c(\boldsymbol{x}_0,\alpha), \qquad I_s(\alpha) = a\eta v_s(\boldsymbol{x}_0,\alpha),$$

där η definierades i uppgift 3 ovan. Visa det första sambandet, om I_c och v_c. Notera att I_c och I_s beror på g men inte på $\boldsymbol{x}_0$. De kan alltså beräknas utan att känna källans läge $\boldsymbol{x}_0$.

Ledning: Använd resultaten i uppgifterna 1, 2 och 3 ovan. Utnyttja informationen om S. Variabelbytet $\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{x}_0$ är användbart, och även additionsformlerna för trigonometriska funktioner.

[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, $\text{supp}(S) := \overline{\{\boldsymbol{x} \in \mathbb{R}^2 \mid S(\boldsymbol{x}) \neq 0\}}$.

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

$$S_0(|\boldsymbol{x}|) = \cos(24|\boldsymbol{x}|)\, e^{-900|\boldsymbol{x}|^2}.$$

Frekvensen är $\omega = 19$ om inget annat sägs. Från analysen ovan vet vi då att

$$I_c(\alpha) = a\eta v_c(\boldsymbol{x}_0, \alpha) = a\eta \cos(\omega(x_0 \cos\alpha + y_0 \sin\alpha)), \qquad \int_{\mathbb{R}^2} S_0(|\boldsymbol{x}|)\cos(\omega x) d\boldsymbol{x} =: \eta. \tag{4}$$

för alla α.

1. Beräkna konstanten η med numerisk integration i två dimensioner. (Ni får inte använda Matlabs inbyggda integrationsmetoder här.)
2. Välj själv styrka och position för källan och beräkna g med `hhsolver.m`. Använd N tillräckligt stort (minst 100). Källan ska ligga en bit in i rummet så att $S \approx 0$ vid väggarna.
3. Sambandet (4) kan användas för att bestämma källans styrka och position. Det gäller för alla α, men i princip räcker det med tre värden på α, vilket ger tre ekvationer, för att bestämma de tre obekanta a, x_0 och y_0. Eftersom både I_c och η kommer innehålla numeriska fel är det emellertid bättre att använda flera α-värden än så och lösa systemet i minstakvadrat-mening. Det ger en robustare metod. Stegen blir:
   - 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 stycken α-värden $\alpha_1, \ldots, \alpha_M$, där M är minst 10.
   - 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

     $$I_c(\alpha_j) \approx \tilde{a}\eta \cos(\omega\tilde{x}_0 \cos\alpha_j + \omega\tilde{y}_0 \sin\alpha_j), \qquad j = 1, \ldots, M.$$

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

   Ledning: För att verifiera att allt blivit rätt med η och $I_c(\alpha)$ kan ni plotta vänsterledet $I_c(\alpha)$ och högerledet $a\eta v_c(\boldsymbol{x}_0, \alpha)$ i (4) som funktion av α, med de kända värdena på x_0, y_0 och a. Det bör ge två identiska kurvor.
4. I praktiken kommer de uppmätta värdena för g, normalderivatan av amplituden, innehålla fel. Lägg därför på brus (normalfördelade slumptal med väntvärde noll) på 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.)

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

$$\frac{\partial v_c(\boldsymbol{x}_0,\alpha)}{\partial \alpha} = \omega(x_0 \sin\alpha - y_0\cos\alpha)v_s(\boldsymbol{x}_0,\alpha) \quad \Rightarrow \quad I_c'(\alpha) = \omega(x_0\sin\alpha - y_0\cos\alpha)I_s(\alpha),$$

$$I_c(\alpha)^2 + I_s(\alpha)^2 = a^2\eta^2(v_c(\boldsymbol{x}_0,\alpha)^2 + v_s(\boldsymbol{x}_0,\alpha)^2) = a^2\eta^2.$$

Genom att tex sätta in $\alpha = 0$ och $\alpha = \pi/2$ ger det, när g är brusfri, att

$$x_0 = \frac{I_c'(\pi/2)}{\omega I_s(\pi/2)}, \qquad y_0 = -\frac{I_c'(0)}{\omega I_s(0)}, \qquad a = \frac{1}{\eta}\sqrt{I_c(0)^2 + I_s(0)^2}.$$

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 3 och 4 ovan. Hur mycket brus klarar algoritmen av nu när man inte antar att källans läge är känt?

Bättre startgissningar: Startgissningarna kan också beräknas från andra värden på α. Fler än två α-värden kan användas; (x_0, y_0) får då minstakvadratanpassas till

$$I_c'(\alpha_j) \approx \omega(x_0\sin\alpha_j - y_0\cos\alpha_j)I_s(\alpha_j), \qquad j = 0, 1, \ldots.$$

Amplituden kan uppskattas med medelvärdet av $\frac{1}{\eta}\sqrt{I_c(\alpha_j)^2 + I_s(\alpha_j)^2}$. När g innehåller brus ger detta robustare värden. Experimentera gärna med denna metod. (Frivilligt!)

6. På hemsidan finns ett antal filer med g för okända isotropa källor av olika svårighetsgrad, med olika frekvenser. Testa att hitta dessa. Källornas g innehåller brus och andra störningar. Bra startgissningar behövs. Beräkna den slutgiltiga residualen i minstakvadratanpassningen för att bedöma hur bra era resultat är. Ni behöver inte klara av alla!

Del 4 – Optimal placering av källa

Du har köpt en TV till rummet och vill sätta upp den på väggen som beskrivs av $\Gamma_{\mathrm{TV}} = [0.6,\ 1] \times 0.6 \subset \partial\Omega$. Du vill hänga upp den så att ljudet i din sovhörna blir så lågt som möjligt i förhållande till ljudet i övriga rummet. Sovhörnan ligger i övre vänstra hörnet av rummet och beskrivs av $\Omega_{\mathrm{sov}} = [0,\ 0.25] \times [0.5,\ 0.75] \subset \bar{\Omega}$.

För att lösa detta antar vi att ljudet från TVn kommer från en isotrop ljudkälla med styrka $a = 1$ av samma typ som i Del 3 ovan. Källan är centrerad i en punkt på Γ_{TV}, dvs

$$S(\boldsymbol{x}) = S_0(x - x_s, y - 0.6),$$

för $x_s \in [0.6,\ 1]$. Vi låter relativa ljudstyrkan i sovhörnet A definieras som kvoten mellan maxnormen av $|u|$ i sovhörnan och i hela rummet,

$$A = \frac{\max_{\boldsymbol{x}\in\Omega_{\mathrm{sov}}} |u(\boldsymbol{x})|}{\max_{\boldsymbol{x}\in\Omega} |u(\boldsymbol{x})|}.$$

Eftersom lösningen u beror på TVns placering x_s kommer även ljudstyrkan göra det, så att $A = A(x_s)$. Vi antar slutligen att frekvensen är $\omega = 30$. (Du har noterat att du huvudsakligen störs av denna frekvens.)

1. Lös problemet för några värden $x_s \in [0.6,\ 1]$ och plotta $A(x_s)$ för att grovlokalisera det globala minimumet. (Det kan finnas flera lokala minimum.) Använd ett relativt litet värde på N för att det inte ska ta för lång tid.

   Ledning: Lösingen ges av `hhsolver()`. Följande kod kan användas för att evaluera A:

   ```
   >> w = find(Sol.x<=0.25 & Sol.y>=0.5);
   >> A = max(abs(Sol.u(w)))/max(abs(Sol.u(:)));
   ```

2. Bestäm den optimala positionen x_s^* och motsvarande ljudnivå $A(x_s^*)$ med ett fel som är mindre än 10^{-4}. Du behöver då använda $N = 1000$ eller större och en effektiv algoritm för att hitta x_s^*. Implementera gyllene snittet-sökning för detta. Provkör den först med mindre N. Rapportera dina värden på x_s^* och $A(x_s^*)$. Visualisera resultaten med lämpliga figurer (tex lösningen, ljudvågorna, för en "bra" och en "dålig" TV-placering).

3. Antag slutligen att du kan ställa TV-högtalaren en bit ifrån väggen, så att TV-ljudet också tillåts komma från en punkt inne i rummet. Detta leder till ett optimeringsproblem i två variabler, då $A = A(x_s, y_s)$. Lös detta med MATLABs `fminsearch()` (som använder Nelder-Meads algoritm). Prova med olika startgissningar till du hittar ett minimum inne i rummet. Hur mycket lägre ljudstyrka kan du uppnå? Visualisera resultatet!

   Ledning: Grovlokalisera först ett minimum med ett litet N. Använd sedan detta som startgissning för större N. För att få feedback under minimeringen kan man använda följande option till `fminsearch()`:

   ```
   >> options = optimset('Display','iter');
   >> Xopt = fminsearch(F, X0, options);
   ```