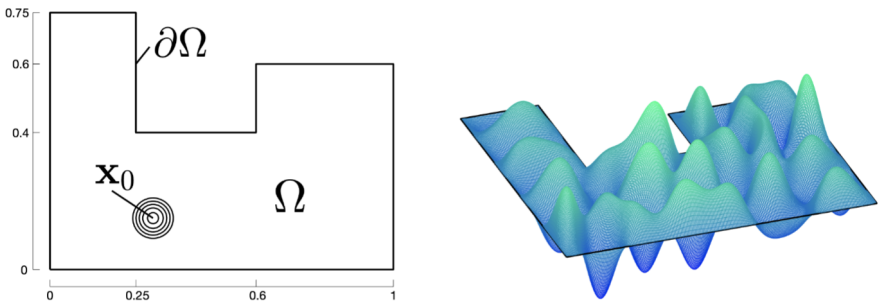
Projekt numeriska metoder

Jim Ogenstad
jim@ogenstad.com

15 maj 2024

Inledning

Detta projekt är ett studium av ljudvågor i ett slutet rum. Mer specifikt betraktas ett problem där positionen av en ljudkälla ska bestämmas utifrån mikrofoner längst väggarna i rummet samt ett problem där en ljudkälla ska placeras för att minimera ljudnivån i ett visst område i rummet. För att exemplifiera hur detta kan se ut visas nedanför rummet och en visuell representation av ljudvågorna.



Figur 1: Till vänster: rummet. Till höger: Ljudvågorna om källan placeras i $(0.3, 0.15)$

Ljud betraktas som bekant med tryckvariationer och i rummet kan vi beskriva denna variation som en funktion $p = p(t, \boldsymbol{x})$ där $\boldsymbol{x} = (x, y)$. Givet att vi har en fix frekvens f kan då trycket skrivas som $p(t, \boldsymbol{x}) = u(\boldsymbol{x})\cos(\omega t)$. Vidare införs en källfunktion $S = S(\boldsymbol{x})$ som beskriver ljudproduktionen givet en punkt i rummet. Värt att notera här är att produktionen är näst intill noll i alla punkter förutom de nära källan $\boldsymbol{x}_0 = (x_0, y_0)$. Sedan kan $u(\boldsymbol{x})$ bestämmas via den

partiella differentialekvationen (Helmholz ekvation):

$$\Delta u(\boldsymbol{x}) + \omega^2 u(\boldsymbol{x}) = S(\boldsymbol{x}), \tag{1}$$

för alla positioner i rummet Ω. På grund av hur ljudet reflekteras vid väggarna kan vi anta $u(\boldsymbol{x}) = 0$ för alla $\boldsymbol{x}$ längst randen av Ω.

Som tidigare nämnts är den första uppgiften att kunna lokalisera en källa i rummet. För att göra detta antar vi att $S(\boldsymbol{x}) = aS_0(\boldsymbol{x} - \boldsymbol{x}_0)$. Funktionen S_0 kommer vara känd och symmetrisk, dvs $S_0(\boldsymbol{x}) = S_0(|\boldsymbol{x}|)$. De obekanta är källans position $\boldsymbol{x}_0$ och styrkan a. För att kunna lokalisera källan används data från mikrofoner som sitter längst väggarna och mäter upp normalderivatan

$$g(\boldsymbol{x}) := \nabla u(\boldsymbol{x}) \cdot \hat{n}(\boldsymbol{x}).$$

För att lösa uppgifterna kunde först samband härledas i för att sedan beräkna en källas position numeriskt. Sedan har vi i det första problemet lokaliserat källans position i källa 1 till 4 som ca $(0.63475, 37489)$, $(0.17754, 0.34714)$, $(0.83607, 0.43673)$ respektive $(0.15633, 0.65123)$. Sedan i det andra problemet om att minimera ljudet i en viss del av rummet användes gyllenesnittet sökning för att hitta den optimala positionen längst väggen i $x = 0.687515774914084$, $y = 0.6$. Vidare hittades den minsta ljudnivån i rummet med källan i punkt $(0.79225, 0.43203)$ med hjälp av matlabs inbyggda minimeringsfunktion fminsearch som utnyttjar Helder-Meads algoritm.

Teoretisk del

Givet vår information om frekvensen ω, funktionen S_0 och normalderivatorna $g(\boldsymbol{x})$ vill vi hitta samband för att kunna beräkna $\boldsymbol{x}_0$ och a.

Steg 1

Vi börjar nu att anta en funktion $v(\boldsymbol{x})$ som uppfyller Helmholz ekvation (1) med högerledet noll. Med detta kan vi utgå från Helmholz ekvation (1) och multiplicera båda leden med $v(\boldsymbol{x})$ och integrera över Ω enligt

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

Vänsterledet kan först delas upp i två termer, sedan kan vi utföra partiell integration på den andra två gånger, vilket ger

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

$$\int_\Omega \omega^2 u(\boldsymbol{x}) v(\boldsymbol{x}) d\boldsymbol{x} + \int_{\partial\Omega} v(\boldsymbol{x}) \nabla u(\boldsymbol{x}) \hat{n} ds - \int_\Omega \nabla v(\boldsymbol{x}) \nabla u(\boldsymbol{x}) d\boldsymbol{x} =$$

$$\int_\Omega \omega^2 u(\boldsymbol{x}) v(\boldsymbol{x}) d\boldsymbol{x} + \int_{\partial\Omega} v(\boldsymbol{x}) \nabla u(\boldsymbol{x}) \hat{n} ds - \int_{\partial\Omega} \nabla v(\boldsymbol{x}) u(\boldsymbol{x}) \hat{n} ds + \int_\Omega \Delta v(\boldsymbol{x}) u(\boldsymbol{x}) d\boldsymbol{x}$$

$$= \int_\Omega u(\boldsymbol{x})(\omega^2 v(\boldsymbol{x}) + \Delta v(\boldsymbol{x})) d\boldsymbol{x} + \int_{\partial\Omega} v(\boldsymbol{x}) \nabla u(\boldsymbol{x}) \hat{n} ds - \int_{\partial\Omega} \nabla v(\boldsymbol{x}) u(\boldsymbol{x}) \hat{n} ds.$$

Vi noterar nu att eftersom funktionen v är definierad enligt $\omega^2 v(\boldsymbol{x}) + \Delta v(\boldsymbol{x}) = 0$ så har den första integralen värdet noll. Sedan ser vi att den tredje integralen är innehåller en faktor $u(\boldsymbol{x})$. Men eftersom denna faktor är noll längst hela randen så är den tredje integralens värde också noll. Kvar blir bara den andra integralen

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

Denna sista likhet ska vi senare se kommer ge information om de sökta okända variablerna.

Steg 2

För att kunna dra nytta av likheten måste vi först hitta en funktion $v(\boldsymbol{x})$ som följer kravet i dess definition. Därför gör vi ansatserna $v_c(\boldsymbol{x}, \alpha) = \cos(\omega(x\cos\alpha + y\sin\alpha))$ och $v_s(\boldsymbol{x}, \alpha) = \sin(\omega(x\cos\alpha + y\sin\alpha))$. Vi vill visa att dessa två satisfierar villkoret $\omega^2 v(\boldsymbol{x}) + \Delta v(\boldsymbol{x}) = 0$. Vi gör detta lämpligen genom att först beräkna $\Delta v(\boldsymbol{x})$, först för v_c. Vi har

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

$$= \nabla(-\omega \cos\alpha \sin(\omega(x\cos\alpha + y\sin\alpha)), -\omega \sin\alpha \sin(\omega(x\cos\alpha + y\sin\alpha))) =$$

$$-\omega^2 \cos^2\alpha \cos(\omega(x\cos\alpha + y\sin\alpha)) - \omega^2 \sin^2\alpha \cos(\omega(x\cos\alpha + y\sin\alpha)) =$$

$$= -\omega^2(\cos^2\alpha + \sin^2\alpha)\cos(\omega(x\cos\alpha + y\sin\alpha)) = -\omega^2 \cos(\omega(x\cos\alpha + y\sin\alpha))$$

$= -\omega^2 v_c(\boldsymbol{x}, \alpha)$. Eftersom $\Delta v_c(\boldsymbol{x}, \alpha) = -\omega^2 v_c(\boldsymbol{x}, \alpha)$ så kan vi direkt se att v_c uppfyller villkoret. Beviset är näst intill identiskt för funktionen v_s. Där kommer vi derivera sinus två gånger och få tillbaka negativa sinus, ω^2 och trigonometriska ettan.

Steg 3

Nu vill vi börja använda dessa funktioner för integreringen, men först vill vi konstantera två samband. Först har vi

$$\int_{\mathbb{R}^2} S_0(|\boldsymbol{x}|) v_s(\boldsymbol{x}, \alpha) d\boldsymbol{x} = 0$$

Detta samband kommer vi inte att bevisa. Det andra sambandet är

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

Vi betecknar denna högra integral som η. Sedan kan vi utgå ifrån vänsterledet för att härleda likheten. Vi har

$$\int_{\mathbb{R}^2} S_0(|\boldsymbol{x}|) v_c(\boldsymbol{x}, \alpha) d\boldsymbol{x} = \int_{\mathbb{R}^2} S_0(|\boldsymbol{x}|) \cos(\omega(x\cos\alpha + y\sin\alpha)) d\boldsymbol{x}.$$

Vi börjar göra ett variabelbyte till polära koordinater enligt $x = \cos\theta$, $y = \sin\theta$, $|J| = r$. Detta ger

$$\int_0^{2\pi}\int_0^{\infty} S_0(r) r\cos(\omega r(\cos\theta\cos\alpha+\sin\theta\sin\alpha)drd\theta =$$

$$= \int_0^{2\pi}\int_0^{\infty} S_0(r) r\cos(\omega r\cos(\theta-\alpha))drd\theta.$$

För att inse att vinkeln α är obetydlig kan vi göra variabelbytet $l = r$, $\varphi = \theta - \alpha$, $|J| = 1$. Detta ger

$$\int_{\alpha}^{2\pi+\alpha}\int_0^{\infty} S_0(l) l\cos(\omega l\cos(\varphi))dld\varphi.$$

Vi inser att det polära koordinatsystemets struktur innebär att $[\alpha, 2\pi+\alpha] = [0, 2\pi]$, därmed kan vi ändra tillbaka gränserna till det senare. Sedan låter vi igen $x = l\cos(\varphi)$, $y = l\sin(\varphi)$, $|J| = 1/l$. Detta ger

$$\int_{\mathbb{R}^2} S_0(|\boldsymbol{x}|)\cos(\omega x)d\boldsymbol{x},$$

vilket skulle visas.

Steg 4

Med detta är vi redo att visa några sista samband. Först betecknar vi integralerna som följer

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

Givet dessa vill vi visa att $I_c(\alpha) = a\eta v_c(\boldsymbol{x}_0,\alpha)$ och $I_s(\alpha) = a\eta v_s(\boldsymbol{x}_0,\alpha)$. Vi nöjer oss med att bevisa det vänstra. Detta samband skulle låta oss beräkna $I_c(\alpha)$ och η numeriskt och sedan helt enkelt lösa ut $\boldsymbol{x}_0$ och a med ett ekvationssystem. Vi kan först skriva ut termerna, vilket ger

$$\int_{\partial\Omega} v_c(\boldsymbol{x},\alpha)g(\boldsymbol{x})ds = av_c(\boldsymbol{x}_0,\alpha)\int_{\mathbb{R}^2} S_0(|\boldsymbol{x}|)v_c(\boldsymbol{x},\alpha)d\boldsymbol{x}$$

Vi börjar med att använda vad vi bevisade i deluppgift 1 genom att skriva om integralen i vänsterledet som

$$\int_{\partial\Omega} v_c(\boldsymbol{x},\alpha)g(\boldsymbol{x})ds = \int_{\Omega} S(\boldsymbol{x})v_c(\boldsymbol{x},\alpha)d\boldsymbol{x}$$

Sen kan vi fortsätta skriva om vänsterledet med $S(\boldsymbol{x}) = aS_0(\boldsymbol{x}-\boldsymbol{x}_0)$. Då får vi

$$\int_{\Omega} S(\boldsymbol{x})v_c(\boldsymbol{x},\alpha)d\boldsymbol{x} = a\int_{\Omega} S_0(\boldsymbol{x}-\boldsymbol{x}_0)v_c(\boldsymbol{x},\alpha)d\boldsymbol{x}.$$

Nu utför vi variabelbytet $\boldsymbol{x}-\boldsymbol{x}_0 = \boldsymbol{y}$ och får

$$a\int_{\Omega} S_0(\boldsymbol{y})v_c(\boldsymbol{y}+\boldsymbol{x}_0,\alpha)d\boldsymbol{y}.$$

Vi kan skriva om $v_c(\boldsymbol{y}+\boldsymbol{x}_0,\alpha)$ som

$$\cos(\omega((y_1+x_{0,1})\cos\alpha + (y_2+x_{0,2})\sin\alpha)) =$$

$$= \cos(\omega(y_1\cos\alpha + y_2\sin\alpha))\cos(\omega(x_{0,1}\cos\alpha + x_{0,2}\sin\alpha)) -$$

$$-\sin(\omega(y_1\cos\alpha + y_2\sin\alpha))\sin(\omega(x_{0,1}\cos\alpha + x_{0,2}\sin\alpha)) =$$

$$= v_c(\boldsymbol{y},\alpha)v_c(\boldsymbol{x}_0,\alpha) - v_s(\boldsymbol{y},\alpha)v_s(\boldsymbol{x}_0,\alpha).$$

Nu kan vi sätta tillbaka detta i integralen och få

$$a\int_{\Omega} S_0(\boldsymbol{y})(v_c(\boldsymbol{y},\alpha)v_c(\boldsymbol{x}_0,\alpha) - v_s(\boldsymbol{y},\alpha)v_s(\boldsymbol{x}_0,\alpha))d\boldsymbol{y} =$$

$$= av_c(\boldsymbol{x}_0,\alpha)\int_{\Omega} S_0(\boldsymbol{y})v_c(\boldsymbol{y},\alpha)d\boldsymbol{y} - av_s(\boldsymbol{x}_0,\alpha)\int_{\Omega} S_0(\boldsymbol{y})v_s(\boldsymbol{y},\alpha)d\boldsymbol{y}.$$

Vi antar att källan ligger en bit ifrån väggarna så att $S(\boldsymbol{y})$ för alla $\boldsymbol{y}$ utanför väggarna är noll. Givet detta kan vi utöka integrationsområdet till $\mathbb{R}^2$ utan att ändra integralernas värde. Alltså har vi nu

$$av_c(\boldsymbol{x}_0,\alpha)\int_{\mathbb{R}^2} S_0(\boldsymbol{y})v_c(\boldsymbol{y},\alpha)d\boldsymbol{y} - av_s(\boldsymbol{x}_0,\alpha)\int_{\mathbb{R}^2} S_0(\boldsymbol{y})v_s(\boldsymbol{y},\alpha)d\boldsymbol{y}$$

Vi vet att $\int_{\mathbb{R}^2} S_0(\boldsymbol{y})v_s(\boldsymbol{y},\alpha)d\boldsymbol{y} = 0$, därför återstår bara den vänstra termen. Om vi går tillbaka till vårt högerled ser vi att vi nu har

$$av_c(\boldsymbol{x}_0,\alpha)\int_{\mathbb{R}^2} S_0(\boldsymbol{y})v_c(\boldsymbol{y},\alpha)d\boldsymbol{y} = av_c(\boldsymbol{x}_0,\alpha)\int_{\mathbb{R}^2} S_0(|\boldsymbol{x}|)v_c(\boldsymbol{x},\alpha)d\boldsymbol{x}$$

Det är lätt att inse att dessa uttryck är lika eftersom vi vet att S_0 enbart beror på dess avstånd till origo och eftersom det inte spelar roll hur man betecknar sin variabel i integralen. Med andra ord är vänster led lika med höger led. Alltså gäller $I_c(\alpha) = a\eta v_c(\boldsymbol{x}_0,\alpha)$.

Lösning av det inversa problemet

Som nämnts tidigare ligger lösningen i sambandet $I_c(\alpha) = a\eta v_c(\boldsymbol{x}_0,\alpha)$. η definierades enligt

$$\eta = \int_{\mathbb{R}^2} S_0(|\boldsymbol{x}|)\cos(\omega x)d\boldsymbol{x}$$

Funktionen S_0 är som sagt känd och nu är den definierad enligt $S_0(|\boldsymbol{x}|) = \cos(20|\boldsymbol{x}|)e^{-1000|\boldsymbol{x}|^2}$. Givet detta beräknas η enkelt med trapetsregeln i två dimensioner. Eftersom S_0 endast är nollskild nära origo så behöver man bara integrera i en liten omgivning däromkring, exempelvis $[-0.2, 0.2] \times [-0.2, 0.2]$. För $\omega = 19$ ger detta $\eta \approx 0.002378$. För beräkningen använde vi $n = 200$ för trapetsregeln i två dimensioner, vilket är så pass högt att en korrekt implementation av trapetsregeln bör ge minimalt fel i svaret. Detta verifierades ytterligare genom att köra pogrammet både med dubbla n, dvs $n = 400$, samt med en mer omfattande omgivning ($[-0.3, 0.3] \times [-0.3, 0.3]$). Detta gav absolut ingen skillnad i det nummeriskt beräknade värdet för η vid användning av det långa formatet i matlab. Av denna anledning kan vi med säkerhet påstå att värdet för η har ett ytterst litet numeriskt fel och kan därför användas utan problem vid vidare beräkningar. Sedan har vi

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

För ett givet α kan vi beräkna denna integral numeriskt med hjälp av Simpsons metod. Vi har nämligen uppmätta värden av normalderivatorna $g(\boldsymbol{x})$ för ett visst antal punkter, samt båglängden från dessa punkter till origo. Eftersom punkterna tycktes vara ekvidistanta kunde Simpsons metod implementeras med h som båglängden för den första punkten efter origo.

Sedan kan vi derivera sambandet $I_c(\alpha) = a\eta v_c(\boldsymbol{x}_0, \alpha)$ med avseende på α vilket ger

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

$$= I_s(\alpha)\omega(x_0 \sin(\alpha) - y_0 \cos(\alpha)).$$

Från detta samband kan man sätta $\alpha = 0$ samt $\alpha = \pi/2$ och lösa ut

$$y_0 = -\frac{I_c'(0)}{\omega I_s(0)}, \;\; x_0 = \frac{I_c'(\pi/2)}{\omega I_s(\pi/2)}.$$

Vi har även

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

$$\iff a = \frac{1}{\eta}\sqrt{I_c(\alpha)^2 + I_s(\alpha)^2}.$$

Dessa värden kunde sedan användas som startvärden för $\boldsymbol{x}_0$ och a vid lösning med hjälp av Gauss-Newtons metod för det ursprungliga sambandet $I_c(\alpha) = a\eta \cos(\omega(x_0 \cos\alpha + y_0 \sin\alpha))$. Där vi får ett system av ekvationer eftersom vi använder fler värden av α. Detta minskar känslighet mot brus i indatan samt numeriska fel. Det finns även risk för överanpassning med för få α värden. Vid implementation av Gauss-Newton blev då funktionen vektorvärd med följande utseende för varje rad j: $F_j = a\eta \cos(\omega(x_0 \cos\alpha_j + y_0 \sin\alpha_j)) - I_c(\alpha_j)$. Vidare blev jacobimatrisen derivatan med avseende på x_0, y_0 respektive a med olika α-värden för varje rad. Specifikt har vi

$$\frac{\partial F_j}{\partial a} = \eta \cos(\omega(x_0 \cos\alpha_j + y_0 \sin\alpha_j))$$

$$\frac{\partial F_j}{\partial x_0} = -a\eta\omega\cos\alpha_j \sin(\omega(x_0\cos\alpha_j + y_0\sin\alpha_j))$$

$$\frac{\partial F_j}{\partial y_0} = -a\eta\omega\sin\alpha_j \sin(\omega(x_0\cos\alpha_j + y_0\sin\alpha))$$

Dessa värden utgör alltså $J_{j,1}, J_{j,2}$ respektive $J_{j,3}$ för varje rad j i jacobimatrisen J. Avbrottsvilkoret var när det största elementet i delta var mindre än 10^{-10}.

Det ska nämnas att det är mycket viktigt att startgissningen är bra i detta problem eftersom det finns många lokala extrempunkter. Av denna anledning räckte det inte med två α värden i 0 och $\pi/2$. Istället behandlades 100 ekvidistanta vinklar mellan 0 och 2π, vilket ledde till ett ekvationssystem på formen $I_c'(\alpha_j) = I_s(\alpha_j)\omega(x_0\sin\alpha_j - y_0\cos\alpha_j)$. Det systemet löstes också med Gauss-Newtons metod. Startgissningen för a blev då medelvärdet av alla a_j. Det ska också sägas att I_c' kunde beräknas numeriskt med derivatans definition med hjälp av

$$I_c'(\alpha) = \frac{I_c(\alpha + h) - I_c(\alpha - h)}{2h},$$

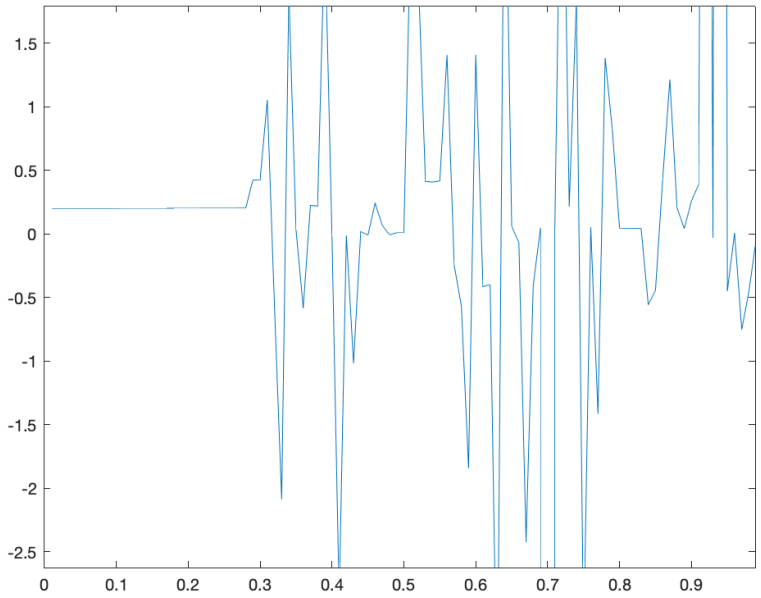
där h valdes till 10^{-6}. För att kontrollera resultaten plottades sedan $I_c(\alpha)$ samt $a\eta v_c(\boldsymbol{x}_0, \alpha)$ som funktioner av α.

Med detta har vi en modell för att läsa in information om normalderivatan längst väggarna och med denna approximera en källas position numeriskt. Ett problem som kan förekomma är en viss nivå av brus. Av denna anledning vill vi se till att modellen faktiskt kan hantera sådant. Det inducerade bruset består av normalfördelade avvikelser i normalderivatorna. Vid testning blir det snabbt tydligt att de slutgiltiga värdena påverkas starkt av startvärdena till Gauss-Newtons metod. Av denna anledning kan vi enkelt hitta en egeninducerad källa trots mycket brus. Specifikt har koden testats med en källa med styrka 1 placerad i punkt $(0.2, 0.2)$. Visserligen har bruset en inverkan på funktionen F i Gauss-Newton eftersom det numeriska värdet $I_c(\alpha_j)$ approximeras under brusets inverkan. Däremot verkar den mer intressanta påverkan av bruset vara på det stora antalet numeriska approximationer som leder till startgissningen.

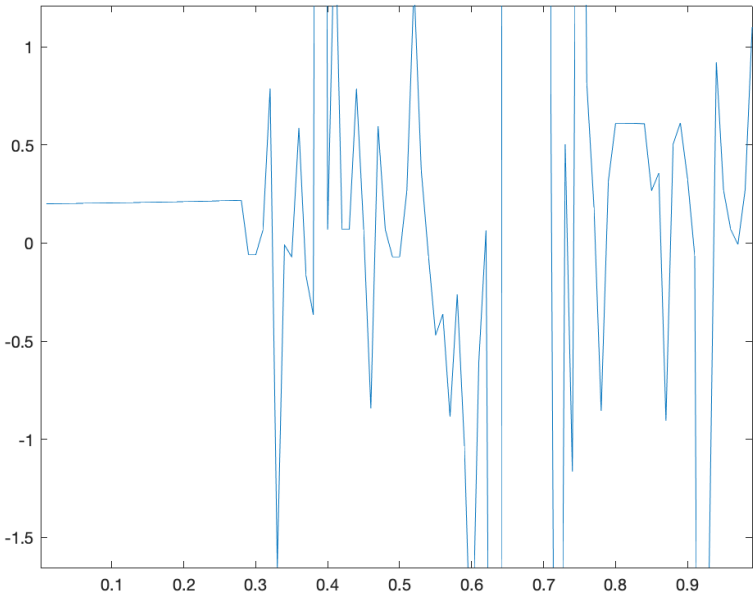
För att exemplifiera kan algoritmen hitta startpositionen med en precision på ca 0.02 i x-led och y-led även med en brusnivå av 1 för vissa körningar givet att startvilkoren i Gauss-Newton matchade precis vad vi genererade. Om vi letar efter en okänd källa kommer vi dock självklart inte redan känna till positionen på förhand, därmed blir det mycket problematiskt om vi bara gissar.

I ett försök att visa detta matades startpoistionen $(0.3, 0.1)$ in i Gauss-Newton, fortfarande med styrkan 1. Då gav algoritmen helt oförutsägbara slutpositioner (fel i storleksordning 10^0 och 10^1) även helt utan brus.

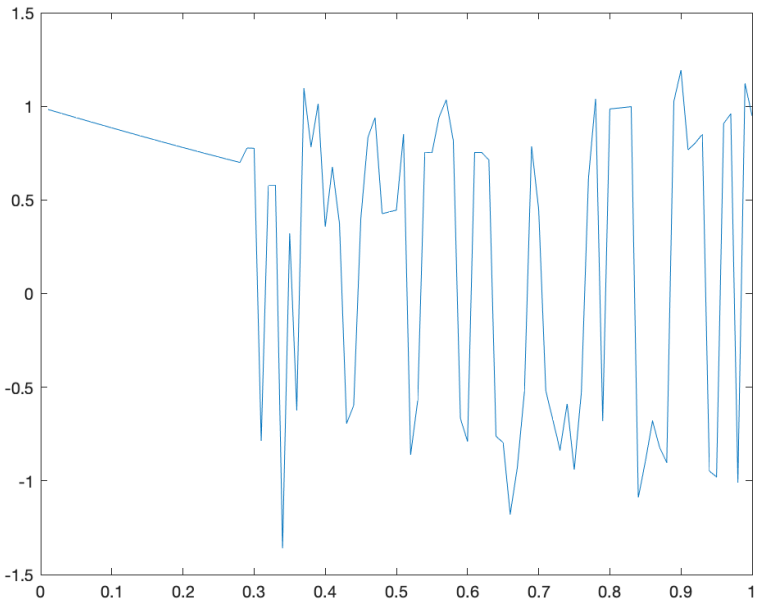
Av denna anledning vänder vi oss till brusets påverkan på algoritmen när startpositionen antas vara okänd och måste approximeras med hjälp av brusdrabbad data. Det är i denna metod vi tydligare kan analysera bruskänsligheten. Ett sätt att till en göra det är att generera ett slumptal för varje värde av normalderivatan och sedan förstora dessa för att påvisa skillnad mellan olika brusnivåer. Anledningen till att inte nya större slumptal generaras är för att det tar väldigt lång tid att köra programmet då. I en körning av programmet ser vi dessa resultat för de slutliga värdena. Datan kommer som sagt från en simmulerad källa med styrka 1 som är placerad i $(0.2, 0.2)$.



Figur 2: Här ser vi $x-$värden för en körning av programmet. Värdena håller sig förhållandevis rimliga fram tills strax innan brusnivå 0.3



Figur 3: Här ser vi $y-$värden för en körning av programmet. Värdena håller sig förhållandevis rimliga fram tills strax innan brusnivå 0.3



Figur 4: Här ser vi $a-$värden för en körning av programmet. Värdena blir gradvis sämre, men runt 0.3 blir bruset för starkt.

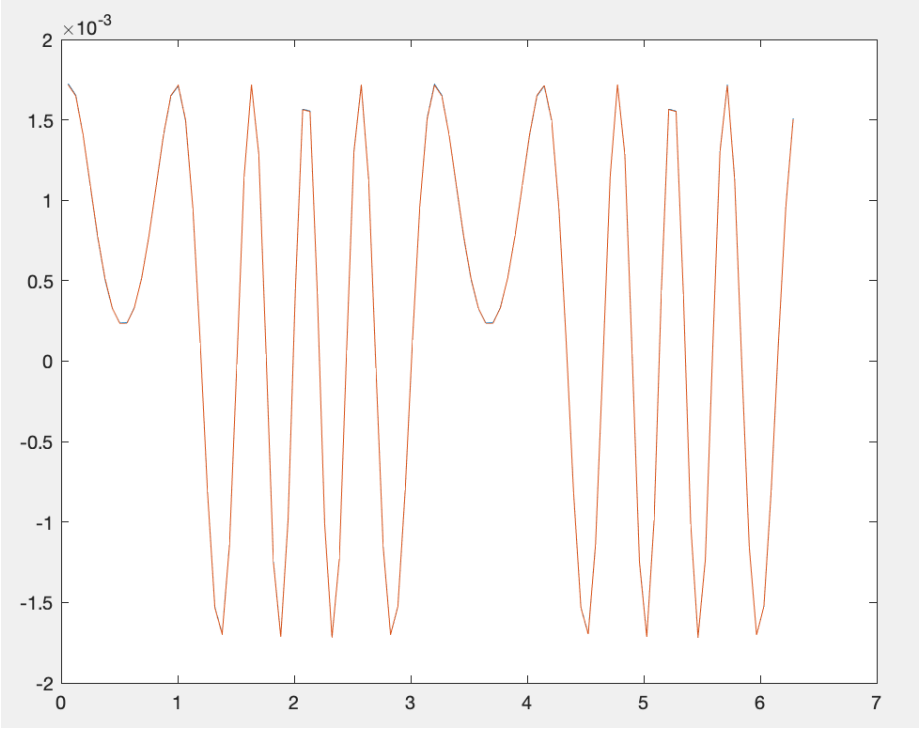
I en körning med brusnivån 0.2 blev startgissningarna till Gauss-Newton $x = 0.09451$, $y = 0.20791$ och $a = 1.0495$. Detta var tillräckligt rimliga startgissningar för att Gauss-Newton skulle konvergera till värden nära den egengenerarade källan. Alltså kan Gauss-Newton hantera vissa fel och fortfarande konvergera rätt, men som sagt blir det svårt om startgissningen väljs helt godtyckligt.

En slutsats som vi kan dra från detta är att algoritmen är förhållandevis brustålig och fungerar relativt bra trots en viss mängd brus. Däremot bör här nämnas igen att vi har samma slumpgenererade tal vid varje körning och att x-axeln bara är en förstoring av de slumpfördelade talen. Av denna anledning har jag kört samma program tio gånger och noterat att den brusnivån som algoritmen tål varierar lite beroende på de slumptal som genereras. I vissa fall sker avbrottet först efter brusnivå 0.5 och i andra fall före 0.2. Det som med säkerhet kan sägas är att algoritmen åtminstone kan hantera en viss nivå av brus och ungefärligt ligger toleransgraden i närheten av 0.3.

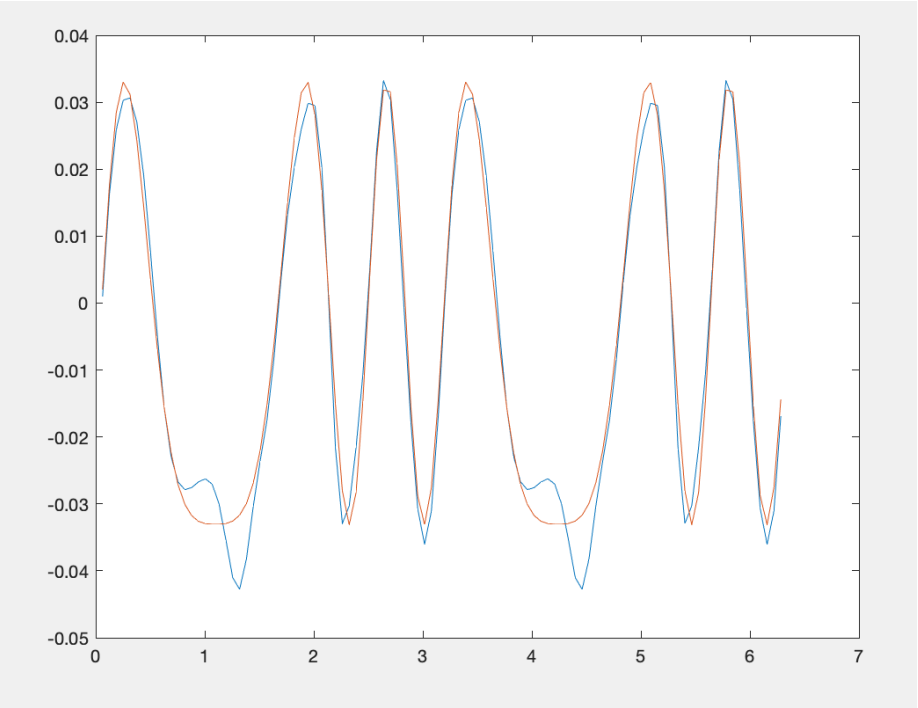
Det bör nämnas att denna analys gjordes av versionen av algoritmen som bara utgår ifrån vinklarna 0 och $\pi/2$ vid bestämning av startgissningar till

Gauss-Newton. Denna metod kunde inte hantera flertalet av de inlästa källorna och behövde därför senare utökas med fler vinklar som beskrivet tidigare.

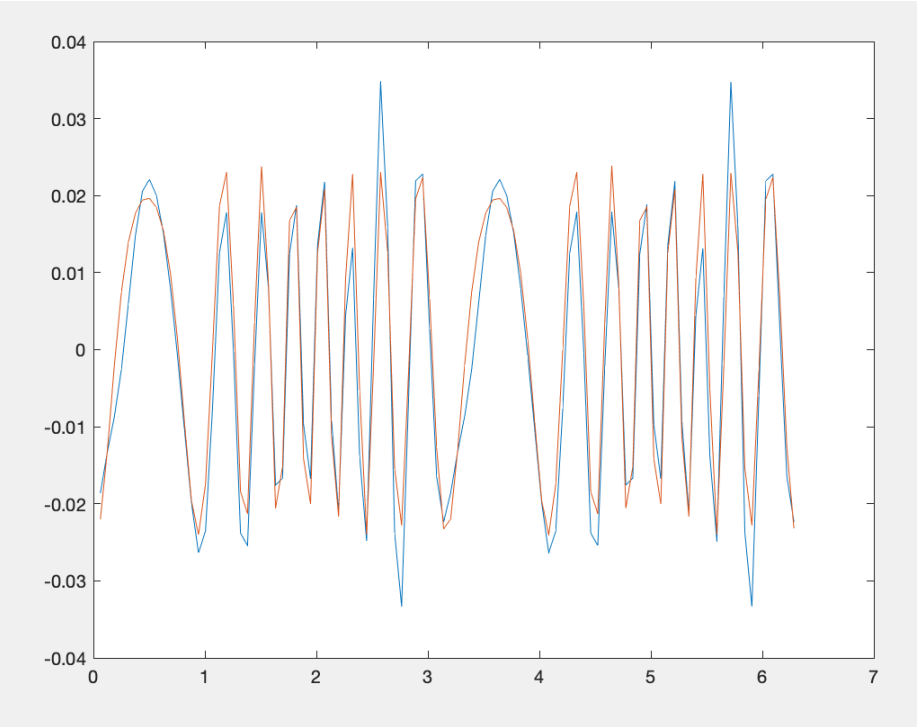
Med informationen om vår metods bruskänslighet kan vi gå vidare till den faktiska inläsningen av ljudkällorna. Här är resultaten för källorna som hittades; nämligen källa 1-4.



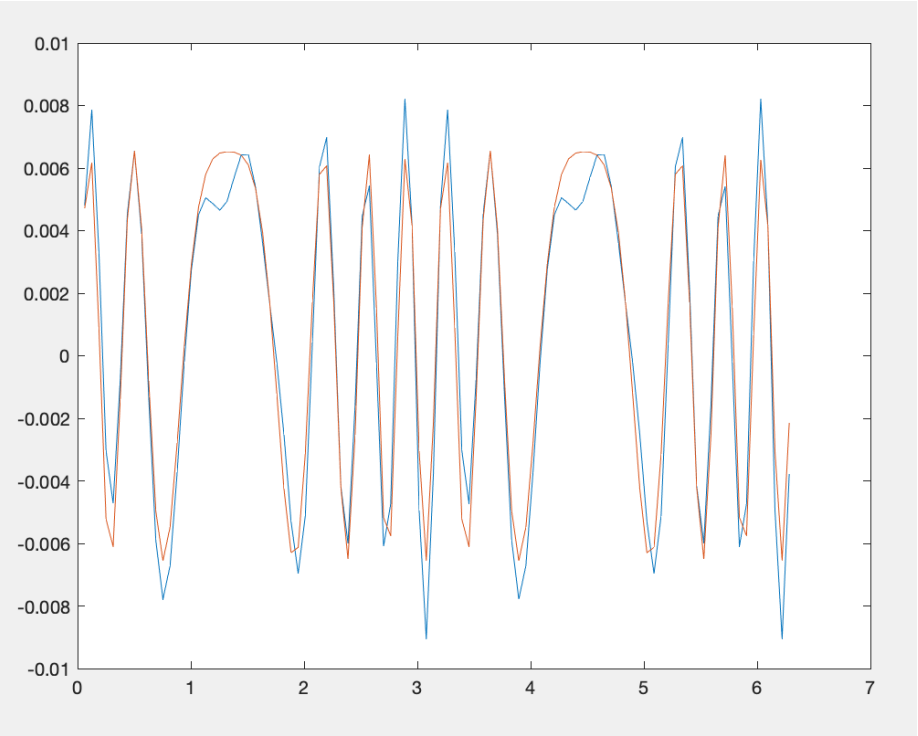
Figur 5: Källa 1. $\omega = 19$. x-axeln motsvarar olika α-värden och på grafen visas $I_c(\alpha)$ i blått och $a\eta v_c(\boldsymbol{x}_0, \alpha)$ i orange. I grafen är $\boldsymbol{x}_0 = (0.63475, 0.37489), a = 0.72388$.



Figur 6: Källa 2. $\omega = 24$ x-axeln motsvarar olika α-värden och på grafen visas $I_c(\alpha)$ i blått och $a\eta v_c(\boldsymbol{x}_0, \alpha)$ i orange. I grafen är $\boldsymbol{x}_0 = (0.17757, 0.34714), a = 14.5321$.

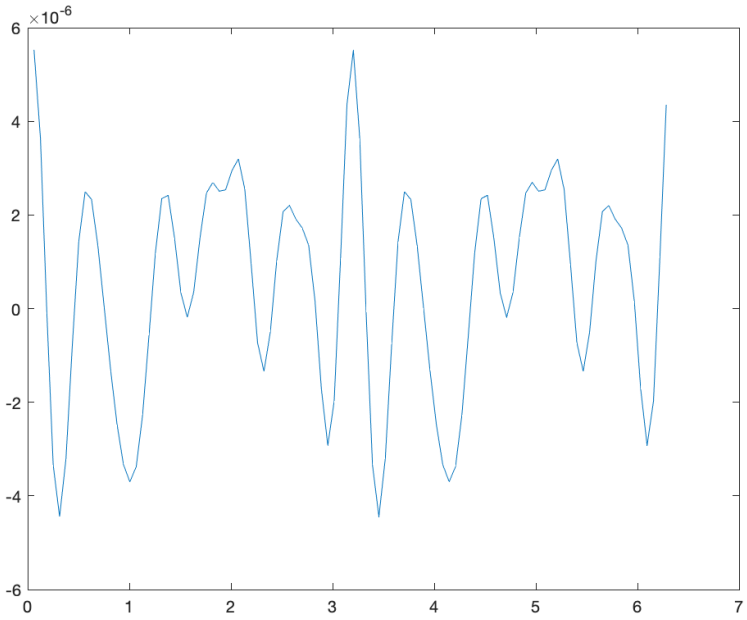


Figur 7: Källa 3. $\omega = 26$ x-axeln motsvarar olika α-värden och på grafen visas $I_c(\alpha)$ i blått och $a\eta v_c(\boldsymbol{x}_0, \alpha)$ i orange. I grafen är $\boldsymbol{x}_0 = (0.83607, 0.43673), a = 10.7455$.

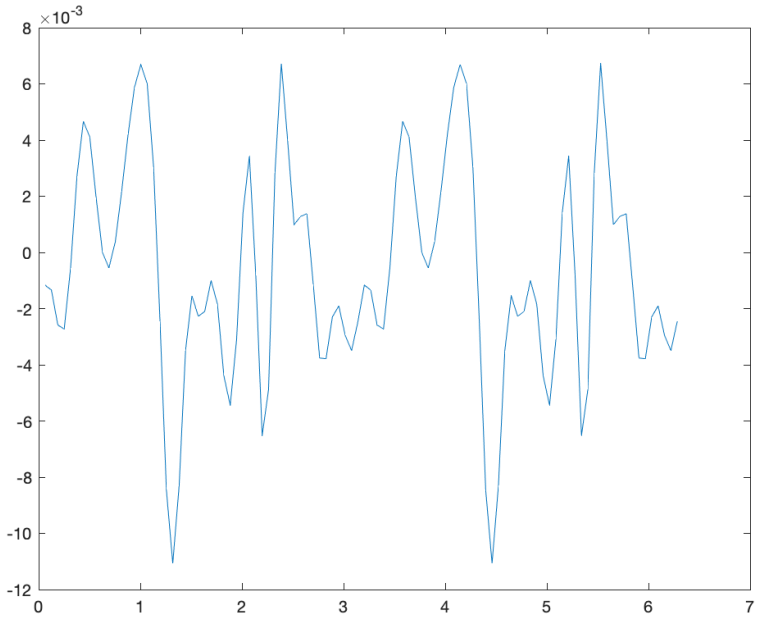


Figur 8: Källa 4. $\omega = 28$ x-axeln motsvarar olika α-värden och på grafen visas $I_c(\alpha)$ i blått och $a\eta v_c(\boldsymbol{x}_0, \alpha)$ i orange. I grafen är $\boldsymbol{x}_0 = (0.15633, 0.65126), a = 2.9967$.

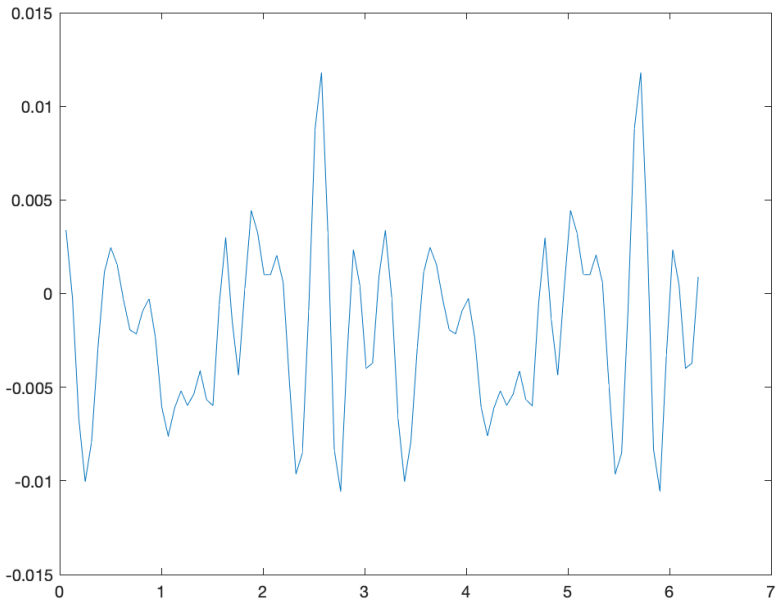
I resultaten ser vi helt klart bäst värden för $\boldsymbol{x}_0$ och a för källa 1, vilket är rimligt eftersom den skulle vara lättast. Mer specifikt kan det bero på olika faktorer såsom låg styrka a, låg frekvens ω samt lågt brus. Vidare kunde positionen till källa 5 inte hittas, vilket kan bero på liknande orsaker. Vi ser rent visuellt att $I_c(\alpha)$ har approximerats relativt bra med $a\eta v_c(\boldsymbol{x}_0, \alpha)$, men vi kan också passa på och plotta differansen mellan den blåa och den orange linjen, dvs residualerna. Återigen har vi olika α värden längst $x-$axeln.



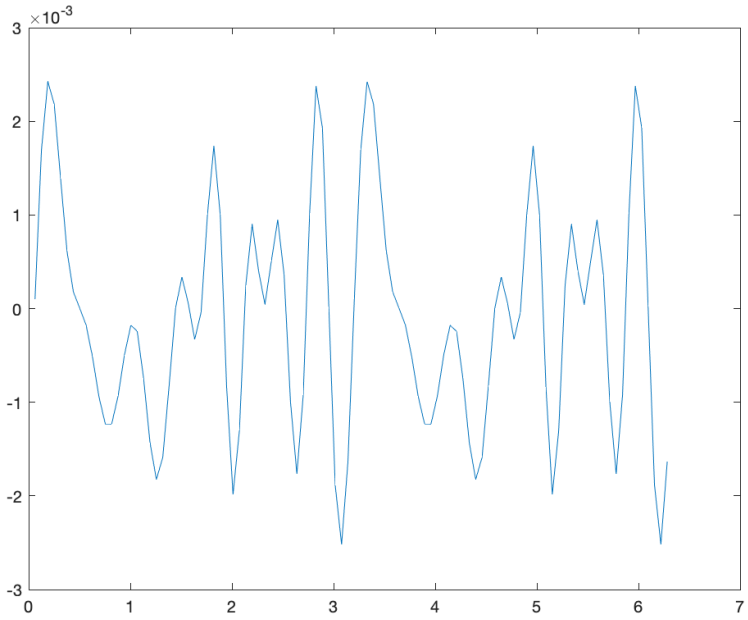
Figur 9: Här ser vi residualerna för källa 1. Tydligen var approximationen bra eftersom skillnaden ligger i storleksordning 10^{-6}.



Figur 10: Här ser vi residualerna för källa 2. Källan har hittats eftersom residualerna endast har storleksordning 10^{-3}.



Figur 11: Här ser vi residualerna för källa 3. Källan har hittats eftersom residualerna endast har storleksordning 10^{-3}.



Figur 12: Här ser vi residualerna för källa 4. Källan har hittats eftersom residualerna endast har storleksordning 10^{-3}.

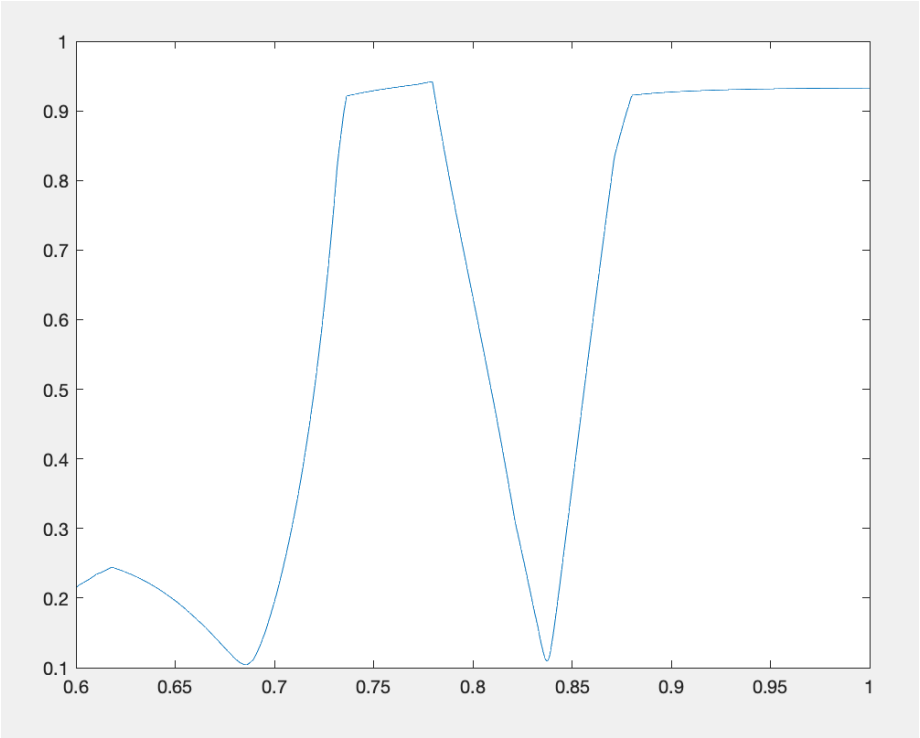
Optimal placering av källa

I det andra problemet finns en sovhörna i $[0, 0.25] \times [0.5, 0.75]$ där ljudet ska minimeras. Detta kan göras med simulationer av ljudvågorna i rummet med hjälp av hhsolver.m. Det som ska minimeras är kvoten

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

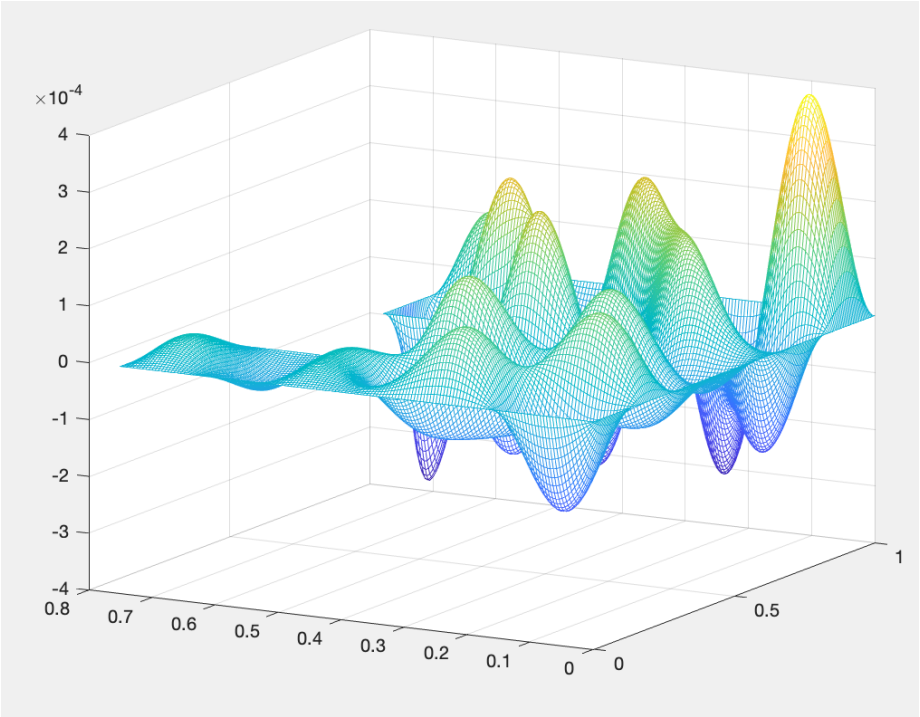
Vid denna minimering utgår vi från frekvensen $\omega = 30$. Först ska positionen optimeras längst väggen $y = 0.6$, $x \in [0.6, 1]$, sedan kan ljudkällan placeras en bit ifrån väggen, vilket leder till ett optimeringsproblem i två variabler.

För den optimala placeringen längst väggen simulerades ljudvågorna 100 gånger, vilket gav följande resultat.



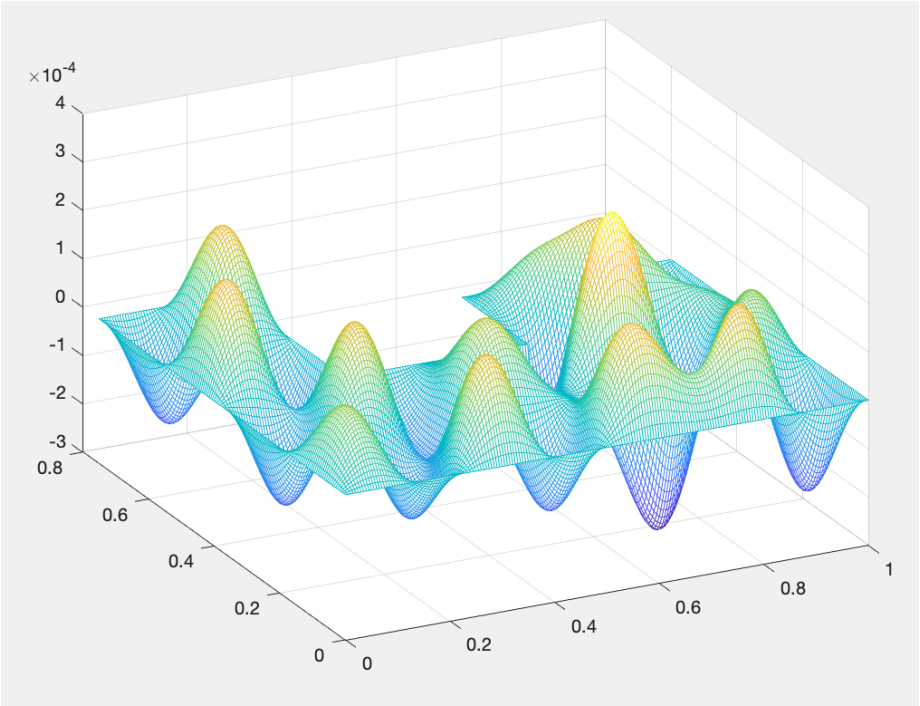
Figur 13: I figuren ser vi olika x-värden längst x-axeln och värdet på kvoten A längst y.

Figuren visar tydligt ett minimum som är mindre än resterande, därför kunde vi anta att funktionen minimeras ungefär mellan 0.65 och 0.70. Sedan användes gyllene snittet-sökning med startpunkter i dessa punkter under antagandet att funktionen endast hade ett minimivärde i intervallet. Koden kör gyllene-snittet tills intervallet är mindre än 10^{-5}, vilket ger en noggrann lösning. Det resulterande x-värdet är 0.687515143648775 och det ger ljudkvoten $A = 0.104855584326738$. Med denna placering av ljudkällan ser ljudvågorna ut på följande vis:



Figur 14: I xy-planet ser vi rummet och i z-led ljudvågorna.

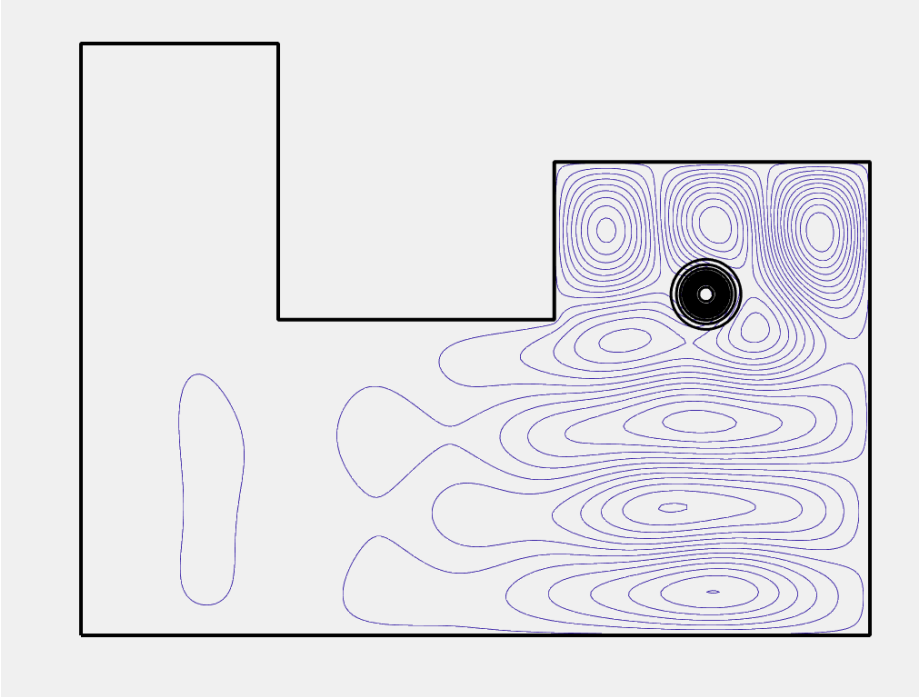
Värt att notera är att sovplatsen är mycket plattare än resten av rummet. För att visa på kontrast är här en bild på hur det ser ut om ljudkällan placeras så att $x = 0.8$



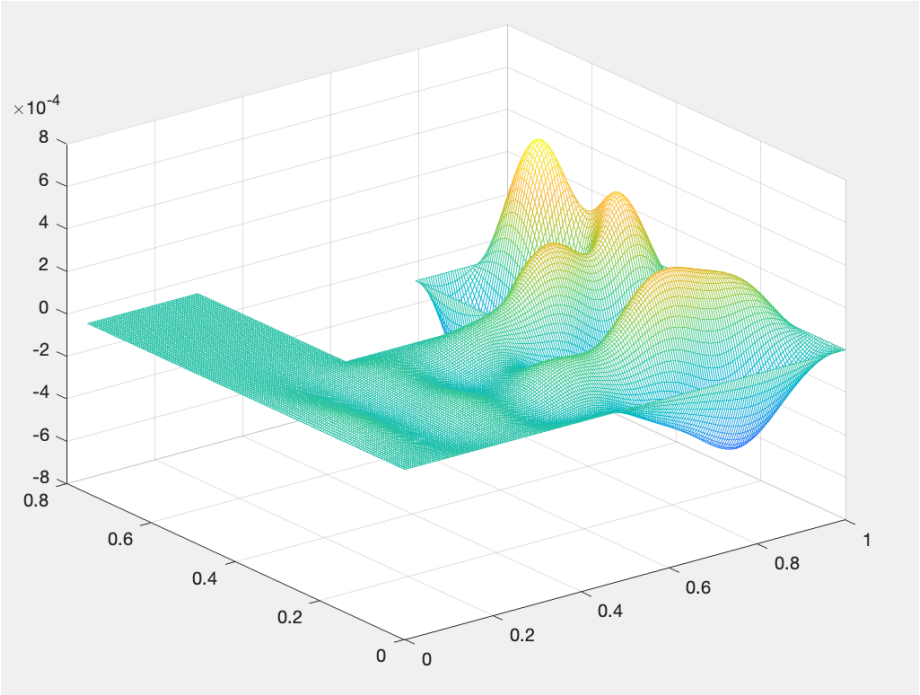
Figur 15: I xy-planet ser vi rummet och i z-led ljudvågorna.

Här ser vi helt klart mer jämnt fördelad ljudnivå i rummet.

Gällande deluppgiften om källans placering i två variabler så användes helt enkelt två nestade slingor i området $[0.6, 1] \times [0.4, 0.6]$ där den inbyggda funktionen fminsearch kördes. Detta gjordes eftersom det fanns väldigt många lokala minpunkter i två dimensioner. Med två nestade slingor hittades många olika minpunkter och den minsta skickades vidare till fminsearch med högre n för hh-solver, vilket gav det minsta värdet $A = 0.00075118$ och den erhölls vid placering av källan i $(0.79225, 0.43203)$. Här är två representationer av de resulterande ljudvågorna.



Figur 16: Här ser vi rummet med ljudkällan i mitten av den svarta cirkeln och ljudvågor illustrerade med svart och lila färg. Vi noterar låg volym i sovhörnan.



Figur 17: I xy-planet ser vi rummet och i z-led ljudvågorna. Här ser vi att det är mycket platt vid sovhörnan.

Det bör noteras att det finns relativt många lokala minpunkter i två dimensioner och att punkternas position inte alltid är helt förutsägbara. Denna variation beror troligen på att vi med exakt precision kan placera ljudkällan så att ljudvågor kancelleras i vissa områden. Med andra ord är lösningarna hyfsat känsliga för fel i positionen, därmed är det svårt att använda de exakta resultaten i verkliga sammanhang.

Egen arbetsinsats

Gällande arbetsinsats har jag skrivit och programmerat allting själv. Som vägledning använde jag mig i princip endast av slides från föreläsningar samt min gamla kod till laborationer, där jag också utgick från slides.