

# SF1693 Labb 1

Jim Ogenstad, Jim@Ogenstad.com

6 december 2024

## Uppgift 1-2 Konvektionsproblem

### Problembeskrivning

Detta problem går ut på att vi modellerar fördelning av rök under verkan av vind i två dimensioner. Vi bildar differentialekvationen  $v(\mathbf{x})\nabla u = f$  där  $v$  motsvarar vindens riktning i en given punkt  $\mathbf{x}$ ,  $u$  är koncentrationen av rök givet en position och den funktion vi önskar att bestämma. Funktionen  $f$  motsvarar rökens produktion för en given punkt. Denna ekvation ska låta oss bestämma rökens koncentration i ett område  $\Omega \subset \mathbb{R}^2$ .

### Enkel analytisk lösning

Som en uppvärmning kan vi lösa problemet för de specifika scenariot  $v(x_1, x_2) = (x_1, x_2)$ ,  $f(x_1, x_2) = 2u(x_1, x_2)$ ,  $\Omega := \{(x_1, x_2) \in \mathbb{R}^2 : x_1 \in \mathbb{R}, x_2 > 1\}$  och med villkoret  $u(x_1, 1) = g(x_1)$ . Differentialekvationen ges då av

$$x_1 \frac{\partial u}{\partial x_1} + x_2 \frac{\partial u}{\partial x_2} = 2u.$$

Vi löser denna analytiskt. Metoden med karakteristikor ger likheten  $\frac{dx_1}{x_1} = \frac{dx_2}{x_2} = \frac{du}{2u}$ . Från det första får vi  $x_1 = x_2 \cdot c$  genom integration av båda leden. Alltså motsvarar  $h(c) = h(x_1/x_2)$  en bana längst karaktäristikan. Där  $h$  är en godtycklig kontinuerlig funktion. Sen har vi likheten  $\frac{dx_1}{x_1} = \frac{du}{2u} \implies u = x_1^2$ . Lösningen ges då av  $u(x_1, x_2) = x_1^2 h(x_1/x_2)$ . Sen ser vi att randvillkoret  $u(x_1, 1) = g(x_1)$  gör att vi kanske vill ha lösningen på en form där vi bättre kan utnyttja  $x_2 = 1$ . Vi kan backa lite och istället använda likheten  $\frac{dx_2}{x_2} = \frac{du}{2u} \implies u = x_2^2$ . Då kan vi istället skriva den allmänna lösningen som  $u(x_1, x_2) = x_2^2 h_2(x_1/x_2)$ . Det är helt motsvarande, men för att få likhet mellan

de olika skrivsätten är  $h \neq h_2$  såklart. Fördelen med detta skrivsätt är att vi får  $u(x_1, 1) = h_2(x_1) = g(x_1)$ , dvs  $g = h_2$ . Därmed kan vi skriva lösningen som  $u(x_1, x_2) = x_2^2 g(x_1/x_2)$  där  $g$  är just funktionen från villkoret vi antog.

## Praktisk tillämpning

I praktiken kommer vi aldrig få vindens riktning, eller produktionen som analytiska uttryck utan istället med hjälp av uppmätt data. Vi har ingen uppmätt data att använda, men vi kan öva på lösningsmetoder genom att själva generera data och bestämma koncentrationen utifrån denna. Rent konkret genererar vi 1000 punkter över delrummet  $\Omega := [0, 1] \times [0, 1]$  likformigt och låter den ”uppmätta” vinden och produktionen ges av funktionerna

$$v(x, y) = (y, 1 - x),$$

$$f(x, y) = \begin{cases} 1, & \text{för } |(x, y) - (0.5, 0.5)| < 0.1 \\ 0, & \text{annars} \end{cases}$$

Relativt enkla funktioner, specifikt ser vi att produktionen sker i mitten av  $\Omega$  och är helt jämn i en cirkel med radie 0.1. Vi kan tillämpa dessa funktioner på våra genererade punkter och få vad vi kan kalla mätvärden för produktionen och vinden i 1000 punkter. Utifrån dessa mätvärden vill vi konstruera approximationer till en analytiska funktioner för  $v_1, v_2$  och  $f$ . Vi gör det med hjälp av en fourierbas. Ett rimligt antagande är att dessa funktioner kan skrivas som fourierserier. Vad vi gör är att vi beskriver funktionen som en sådan, fast med ett ändligt antal termer  $\exp(2\pi i(n_1 x + n_2 y))$ . Vi kan göra det för alla  $n_1, n_2 \in [-N, N]$ . I MATLAB implementeringen låter vi  $N = 8$ . En fourier-summa av denna typ bör approximera funktionerna bra eftersom fourierserier konvergerar. Vi behöver bara ett bra sätt att bestämma koefficienterna framför baserna. Kort sagt gör vi det genom minstakvadratmetoden. I grund och botten är metoden identisk för att bestämma,  $v_1, v_2$  och  $f$ . Vi visar hur det fungerar för att bestämma  $v_1$  och menar att  $v_2$  och  $f$  endast skiljer sig i notation.

Konceptet är att vi vill välja fourierkoefficienterna så bra som möjligt. För varje genererad punkt  $(x_p, y_p)$  vill vi få

$$v_1(x_p, y_p) = \sum_{|n_2| \leq N} \sum_{|n_1| \leq N} \hat{v}_1(n_1, n_2) \exp(2\pi i(n_1 x_p + n_2 y_p)).$$

$\hat{v}_1(n_1, n_2)$  betecknar koefficienten till basvektorn relaterad till  $(n_1, n_2)$ . Det är dessa vi vill bestämma. Om vi ställer upp denna likhet för alla 1000 punkter

får vi ett överbestämt system med 1000 likheter och  $(2N + 1)^2 = 289$  okända. Vi beskriver systemet med  $A\mathbf{x} = \mathbf{b}$ . Vi skapar vektorn  $\mathbf{b}$  med  $v_1(x_p, y_p)$  för alla 1000 punkter  $p$ .  $\mathbf{x}$  motsvarar givetvis koefficienterna  $\hat{v}_1(n_1, n_2)$ , de vi vill bestämma. Sedan får vi matrisen  $A$  av  $\exp(2\pi i(n_1 x_p + n_2 y_p))$  där vi varierar punkterna  $(x_p, y_p)$  längst kolumnerna, dvs en punkt för varje rad, och vi varierar valet av  $n_1$  och  $n_2$  längst raderna. Sedan löser vi helt enkelt systemet i minsta kvadrat mening genom  $A^T A\mathbf{x} = A^T \mathbf{b}$ . Allt detta är rätt smidigt i MATLAB. Detta kvadratiska system löser vi bara med solve-kommandot. Det ger den optimala approximationen skriven som en fourierserie med hjälp av datapunkterna. Som sagt använder vi identiska metoder för att hitta koefficienterna till  $v_2$  och  $f$ . Med detta har vi nu möjligheten att för en given punkt approximativt bestämma vindens riktning och värdet av produktionen.

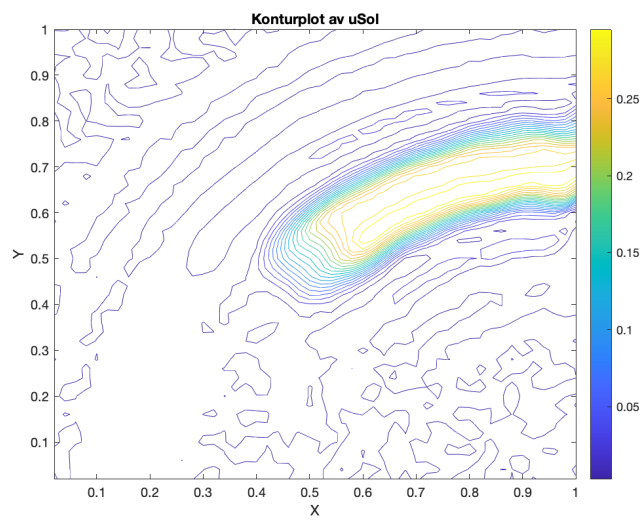
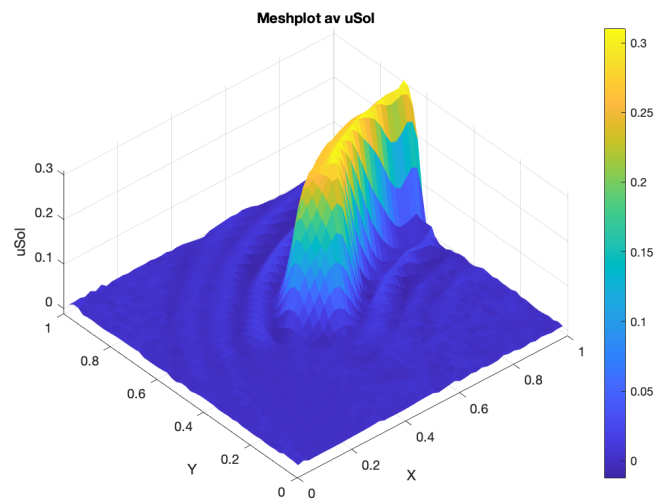
Vi kallar dessa approximerade funktioner för  $\tilde{v}(x, y) = (\tilde{v}_1(x, y), \tilde{v}_2(x, y))$  och  $\tilde{f}(x)$ . Vi vill nu lösa differentialekvationen  $\tilde{v}(x, y)\nabla u(x, y) = \tilde{f}(x, y)$  för  $(x, y) \in \Omega$ . Eftersom viden alltid är positiv kan vi anta att vi i princip har koncentration noll i  $x = 0$  och  $y = 0$  (produktionens centrum var som sagt i  $(0.5, 0.5)$ ). Detta utgör därför randvillkoret. Nästa steg är att lösa differentialekvationen numeriskt. Vi vet hur  $u$  förändrar sig längst en karaktäristika, dvs i vindens riktning. Därav kan vi ta små steg från randen längst vinden och beräkna  $u$  genom Eulers metod. Med liknande tillvägagångssätt kan vi välja en godtycklig punkt på  $\Omega$  och backa i vindens riktning med Eulers metod längst  $x$  och längst  $y$ . Konkret har vi formeln

$$\frac{d}{ds}x(s) = \tilde{v}(\mathbf{x}(s)) \implies \mathbf{x}(s + \Delta) = \mathbf{x}(s) + \Delta\tilde{v}(\mathbf{x}(s)).$$

För något förhållandevis lågt  $\Delta$ . Vi kan välja ett negativt  $\Delta$  för att få en bana  $\mathbf{x}(s)$  som träffar en av koordinataxlarna. Det är precis vad vi vill eftersom det är där vi känner till värdet av  $u$ , nämligen noll (enligt vårt antagande). Detta är bra eftersom vi nu kan beskriva en approximation av vindens bana och eftersom vi känner till hur  $u$  förändras längst denna har vi

$$\frac{d}{ds}u(s) = \tilde{f}(x(s)) \implies u(s + \Delta) = u(s) + \Delta\tilde{f}(x(s)).$$

Vi kan då röra oss från en godtycklig punkt i  $\Omega$  mot randen med små steg. Vid varje steg beräknar vi  $\tilde{f}$  och multiplicerar med  $\Delta$ . Summerar vi detta hela vägen till randen får vi en approximation av  $u$  i punkten vi startade med. Eftersom vi har mätt upp alla skillnader mellan punkter från randen till startpunkten och  $u$  är noll vid kanten så är  $u$  helt enkelt summan av alla skillnader vi adderar ihop. Med denna metod kan vi bestämma ett värde för  $u$  för varje given punkt i  $\Omega$ . I vår MATLAB kod har vi gjort allt detta och hittat lösningar i ett  $50 \times 50$  rutnät i  $\Omega$ , här är resultaten plottade:



## Uppgift 3 Diffusion och slumpvandring

Vi går nu vidare till att studera konvektions-diffusionsproblemet

$$\frac{\partial \rho}{\partial t}(y, t) + \operatorname{div}(v(y)\rho(y, t)) - c\Delta\rho(y, t) = 0$$

$$\rho(y, 0) = g(y) \geq 0,$$

för positiva  $t$  och  $y \in \mathbb{R}^2$ . Specifikt vill vi lösa en förenkling av detta problem där vi låter  $y$  vara endimensionell och vinden  $v$  konstant noll. Syftet är att vi ska jämföra approximation av Fourierserier och finita differensmetoden för lösning av ett diffusionsproblem. Konkret har vi följande differentialekvation:

$$\frac{\partial}{\partial t}u(x, t) - \frac{\partial^2}{\partial x^2}u(x, t) = 0.$$

Denna beskriver en koncentration eller en täthet på intervallet  $x \in [0, \pi]$  och  $t > 0$ . Koncentrationen är noll på randen och initialvilkoret är  $u(x, 0) = g(x)$ . Vi bestämmer lösningar för  $g(x) = x(\pi - x)$  och för

$$g(x) = \begin{cases} x, & x \in [0, \pi/2), \\ 0, & x \in [\pi/2, \pi]. \end{cases}$$

För båda valen av  $g(x)$  löser approximerar vi en lösning med Fourierserier och finita differensmetoden.

### Lösning med Fourierserier

Vi börjar med att lösa differentialekvationen med en konvergent Fourierserie. Vi gör det genom variabelseparation. En lösning antas kunna skrivas som  $u(x, t) = X(x)T(t)$ . Derivering av  $u$  med avseende på variablerna gör att vi kan skriva om ekvationen som

$$T'(t)X(x) = T(t)X''(x) \iff \frac{T'(t)}{T(t)} = \frac{X''(x)}{X(x)}.$$

Utifrån vänster led ser vi att detta uttryck är oberoende av  $x$  och från höger led ser vi oberoende av  $t$ . Därav är uttrycket konstant, vi skriver konstanten som  $-\lambda$  för något  $\lambda \in \mathbb{R}$ . Detta ger två differentialekvationer där båda är av en variabel. För  $X$  har vi

$$X''(x) + \lambda X(x) = 0.$$

Lösningen får olika form beroende på om  $\lambda$  är positiv, negativ eller noll. Vi tittar på fallet  $\lambda = 0$  som ger information om att andraderivatan är noll, alltså är  $X$  linjär. Vi skriver  $X(x) = a + bx$ . Vi vet att  $u$  är noll vid randen, därmed är  $a = 0$ . Om  $u$  även är noll för  $x = \pi$ , oavsett tidpunkt så är även  $b = 0$ , vilket ger  $X(x) = 0$ , alltså inte en särskilt intressant lösning. Vi tittar vidare på fallet då  $\lambda$  är negativ. Det ger lösningen

$$X(x) = c_1 e^{x\sqrt{-\lambda}} + c_2 e^{-x\sqrt{-\lambda}}$$

För att denna ska uppfylla villkoret  $u(0, t) = 0$  gäller nödvändigtvis  $c_1 + c_2 = 0 \iff c_2 = -c_1$ . Vi skriver då om funktionen till

$$X(x) = c_1 \left( e^{x\sqrt{-\lambda}} - e^{-x\sqrt{-\lambda}} \right).$$

Vi ser att  $X(\pi)$  inte är noll förutom om  $\lambda = 0$ , vilket går emot antagandet, eller  $c_1 = 0$ , vilket igen ger en trivial lösning. Därmed är vi klara med detta fall. Om nu  $\lambda > 0$  får vi lösningen

$$X(x) = c_1 \cos(x\sqrt{\lambda}) + c_2 \sin(x\sqrt{\lambda})$$

Dessa koefficienter är alltså inte samma som de vi använde tidigare. Utifrån randvillkoret  $u(0, t) = 0$  ser vi att koefficienten framför cosinus nödvändigtvis är noll. Sedan ser vi  $u(\pi, t) = 0$ , vilket innebär att vi antingen har den triviala lösningen  $c_2 = 0$  eller  $\sin(\pi\sqrt{\lambda}) = 0$ . Den andra likheten löses för alla  $\sqrt{\lambda} = n$  där  $n \in \mathbb{N}$ . Om  $n = 0$  får vi dock tillbaka den triviala lösningen, så vi betraktar istället positiva heltal. Vi vet då att vi har en lösning  $\lambda = n^2$ , för ett positivt heltal  $n$ , med detta löser vi den ordinära ekvationen för  $T$ . Vi får enkelt lösningen genom

$$T(t) = c_3 e^{-tn^2}$$

Det ger en stor uppsättning lösningar och den allmänna lösningen är en linjärkombination av dessa, nämligen

$$u(x, t) = \sum_{n=1}^{\infty} a_n e^{-tn^2} \sin(nx)$$

De specifika värdena  $a_n$  beror på begynnelsevillkoret  $u(x, 0)$ , men vi kan ta fram uttryck för  $a_n$  i det allmänna fallet. Vi har  $u(x, 0) = \sum_{n=1}^{\infty} a_n \sin(nx)$ . Vi multiplicerar med  $\sin(kx)$  och integrerar båda leden över intervallet. Det ger likheten

$$\int_0^{\pi} u(x, 0) \sin(kx) dx = \int_0^{\pi} \sum_{n=1}^{\infty} a_n \sin(nx) \sin(kx) dx.$$

Vi utgår från höger led och börjar med att byta plats på summan och integralen. Vi behöver integrera

$$\int_0^{\infty} \sin(nx) \sin(kx) dx$$

för olika positiva heltal  $n$ . Vi kommer se att vi har ortogonalitet för sinusfunktioner med olika val av  $n$  och  $k$ . Om  $n \neq k$  har vi

$$\int_0^{\infty} \sin(nx) \sin(kx) dx = \int_0^{\pi} \frac{\cos(x(n-k)) - \cos(x(n+k))}{2} dx =$$

$$\left[ \frac{\sin(x(n-k))}{2(n-k)} - \frac{\sin(x(n+k))}{2(n+k)} \right]_0^{\pi} = 0$$

Alla termer blir noll eftersom  $\sin(mx) = 0$  för heltal  $m$ . Därmed blir alla termer utom en i summan noll. Den slutliga termen är

$$a_k \int_0^{\pi} \sin^2(kx) dx = a_k \int_0^{\pi} \frac{1 - \cos(2kx)}{2} dx = a_k \left[ \frac{x}{2} - \frac{\sin(2kx)}{4k} \right]_0^{\pi}$$

$= a_k \frac{\pi}{2}$ . Jämför vi det med vänster led kan vi isolera  $a_k$  för att få

$$a_k = \frac{2}{\pi} \int_0^{\pi} u(x, 0) \sin(kx) dx.$$

Detta låter oss entydigt bestämma Fourierkoefficienterna utifrån ett initialvilkor  $u(x, 0) = g(x)$ . Vi vill som sagt analysera två olika initialvilkor och det första var  $u(x, 0) = g(x) = x(\pi - x)$ . Vi bestämmer Fourierkoefficienterna till detta fall analytiskt genom integrering. Vi går tillbaka till att indexera med  $n$ . Vi har integralen

$$\int_0^{\pi} x(\pi - x) \sin(nx) dx = \pi \int_0^{\pi} x \sin(nx) dx - \int_0^{\pi} x^2 \sin(nx) dx.$$

Vi löser båda integralerna separat och använder partiell integration

$$\begin{aligned} \int_0^{\pi} x^2 \sin(nx) dx &= \left[ -\frac{x^2 \cos(nx)}{n} \right]_0^{\pi} + \int_0^{\pi} \frac{2x \cos(nx)}{n} dx = \\ &= -\frac{\pi^2 \cos(n\pi)}{n} + \left[ \frac{2x \sin(nx)}{n^2} \right]_0^{\pi} - \int_0^{\pi} \frac{2 \sin(nx)}{n^2} dx = \\ &= -\frac{\pi^2 (-1)^n}{n} + \left[ \frac{2 \cos(nx)}{n^3} \right]_0^{\pi} = -\frac{\pi^2 (-1)^n}{n} + \frac{2((-1)^n - 1)}{n^3}. \end{aligned}$$

Vi använder här att  $\sin(n\pi) = 0$  för alla heltal  $n$  och att  $\cos(n\pi) = (-1)^n$ , för heltal  $n$ . Vi beräknar den andra integralen

$$\begin{aligned} \pi \int_0^{\pi} x \sin(nx) dx &= \pi \left[ -\frac{x \cos(nx)}{n} \right]_0^{\pi} + \pi \int_0^{\pi} \frac{\cos(nx)}{n} dx = \\ &= -\frac{\pi^2 (-1)^n}{n} + \pi \left[ \frac{\sin(nx)}{n^2} \right]_0^{\pi} = -\frac{\pi^2 (-1)^n}{n} \end{aligned}$$



Vi subtraherar den första integralen från den andra för att få tillbaka vad vi började med, det ger

$$-\frac{\pi^2(-1)^n}{n} - \left( -\frac{\pi^2(-1)^n}{n} + \frac{2((-1)^n - 1)}{n^3} \right) = -\frac{2((-1)^n - 1)}{n^3}.$$

Vi har därmed  $a_n$  enligt

$$a_n = -\frac{2}{\pi} \frac{2((-1)^n - 1)}{n^3} = -\frac{4((-1)^n - 1)}{\pi n^3}.$$

Med en Fouriersserie kan vi nu lösa problemet genom att stoppa in värdena för  $a_n$ . Först noterar vi dock att  $a_n = 0$  för jämna  $n$  och för udda  $n$  har vi  $a_n = 8/(\pi n^3)$ . Vi stoppar in det i vårt tidigare resultat och får

$$u(x, t) = \sum_{n=1}^{\infty} a_n e^{-tn^2} \sin(nx) = \sum_{n=1}^{\infty} \frac{8}{\pi(2n-1)^3} e^{-t(2n-1)^2} \sin((2n-1)x).$$

Vi summerar bara över alla udda heltal eftersom vi får noll annars. Med detta analytiska uttryck för funktionen kan vi väldigt praktiskt approximera den genom att välja ett ändligt antal termer  $N$  att summera över. Vi kan redan nu kommentera över noggrannheten lite. Vi ser att termerna i summan försvinner med en faktor  $1/n^3$  och med detta kan vi uppskatta hur snabbt vi når en exakt lösning. Har vi exempelvis en approximativ lösning där vi summerar  $N$  termer kommer felet vara summan från  $N+1$  till  $\infty$ . Vi beräknar den summan ungefärligt enligt

$$\sum_{n=N+1}^{\infty} \frac{1}{n^3} \simeq \int_{N+1}^{\infty} \frac{1}{n^3} dn \simeq \frac{1}{N^2}.$$

Felet avtar alltså enligt  $1/N^2$  när vi ökar  $N$ . Vi plottar en lösning med denna metod längre ner, först bestämmer vi koefficienterna  $a_n$  i fallet då

$$g(x) = \begin{cases} x, & x \in [0, \pi/2), \\ 0, & x \in [\pi/2, \pi]. \end{cases}$$

Vi beräknar

$$a_n = \frac{2}{\pi} \int_0^{\pi} g(x) \sin(nx) dx = \frac{2}{\pi} \int_0^{\pi/2} x \sin(nx) dx =$$

$$\frac{2}{\pi} \left[ -\frac{x \cos(nx)}{n} \right]_0^{\pi/2} + \frac{\pi}{2} \int_0^{\pi/2} \frac{\cos(nx)}{n} dx = -\frac{\cos(n\pi/2)}{n} +$$

$$+ \frac{2}{\pi} \left[ \frac{\sin(nx)}{n^2} \right]_0^{\pi/2} = -\frac{\cos(n\pi/2)}{n} + \frac{2 \sin(n\pi/2)}{\pi n^2}.$$

Detta är lite svårare att förenkla eftersom sinus och cosinus ger tre olika svar för multiplar av  $\pi/2$ . Oavsätt kan vi nu skriva  $u$  som

$$u(x, t) = \sum_{n=1}^{\infty} \left( \frac{2 \sin(n\pi/2)}{\pi n^2} - \frac{\cos(n\pi/2)}{n} \right) e^{-tn^2} \sin(nx).$$

Det ser lite krångligt ut, men det är inga problem att implementera detta i matlab. Vi gör på samma sätt som tidigare och ser på termen som krymper långsammast för att bestämma felet. Vi har då felet

$$\left| \sum_{n=N+1}^{\infty} \left( \frac{2 \sin(n\pi/2)}{\pi n^2} - \frac{\cos(n\pi/2)}{n} \right) e^{-tn^2} \sin(nx) \right| \simeq \left| \sum_{n=N+1}^{\infty} \frac{1}{n} \right|.$$

Vi kan säga att vi får konvergens  $O(1)$ , men det betyder egentligen att summan är divergent, alltså försvinner felet inte. Vi har inte punktvis konvergens för alla punkter. Däremot noterar vi att vi har gjort en grov förenkling av felet. Om  $t > 0$  så får vi en extra term  $e^{-tn^2}$  som får summan att konvergera istället, dvs approximationen konvergerar punktvis för  $t > 0$ . Det finns också en viktig skillnad mellan fallet då  $x$  är  $\pi/2$  och fallet då det inte är det. Om  $x = \pi/2$  har vi genom  $\sin(nx)/n$  ett positivt bidrag för varje  $n \equiv 1(\text{mod}4)$  och ett negativt för varje  $n \equiv 3(\text{mod}4)$ . Det ger ett tydligt tyngre bidrag till additionen än till subtraktionen, alltså divergerar serien precis som summan av  $1/n$  gör. Däremot om vi tar  $x$  mer skilt från  $\pi/2$  får vi mer jämnt distribuerade värden eftersom  $\sin(nx)$  varierar någorlunda jämnt för  $x \neq \pi/2$ , vilket resulterar i en kancellering av termer där summan istället konvergerar (detta är iallafall vad resultaten antyder), vi ser senare i plottar att denna konvergens motsvarar  $O(1/\sqrt{n})$ . Alltså har vi punktvis konvergens i alla punkter förutom  $t = 0$ ,

$x = \pi/2$ . Detta är helt rimligt eftersom det följer Gibbs fenomen. Diskontinuiteten i  $g$  vid  $x = \pi/2$  leder till att Fourierserien inte konvergerar punktvis i motsvarande punkt.

Även om vi inte alltid har punktvis konvergens ser vi utan problem att vi kan summera alla termer i fourierserien i kvadrat och få något mindre än  $\sum |1/n^2|$ , vilket konvergerar. Alltså har vi fortfarande  $L^2$  konvergens. Vad vi kan säga på förhand är alltså att Fourierserie approximationen kommer fungera klart bättre med  $g(x) = x(\pi - x)$  än med detta styckvis kontinuerliga begynnelsevillkor.

## Lösning med finita differensmetoden

Vi vill som sagt jämföra denna metod med den finita differensmetoden. Därför börjar vi med att approximera derivatorna med differenskvoter. Först kan vi diskretisera  $u$  över ett rutnät. Vi skriver  $x = n\Delta x$  och  $t = m\Delta t$  för heltal  $n$  och  $m$ . Både  $x$  och  $t$  består alltså av en summa med små steg  $\Delta x$  respektive  $\Delta t$ . Med detta kan vi skriva  $u$  som  $u(x, t) = u(n\Delta x, m\Delta t) = u_{nm}$ . Denna diskretisering gör det praktiskt att approximera derivator. Vi låter nämligen

$$\frac{\partial}{\partial t} u(x, t) \simeq \frac{u_{n(m+1)} - u_{nm}}{\Delta t}$$

$$\frac{\partial^2}{\partial x^2} u(x, t) \simeq \frac{u_{(n+1)m} - 2u_{nm} + u_{(n-1)m}}{\Delta x^2}.$$

Eftersom vi känner till begynnelsevillkoret  $u(x, 0)$  så vill vi rekursivt kunna bestämma  $u$  vid en tidpunkt, givet att vi känner till funktionens värde vid tidpunkten innan. Vi gör detta genom att skriva ut differentialekvationen approximerad med differenskvoter och sen isolerar vi den senare tidpunkten

$$\frac{u_{n(m+1)} - u_{nm}}{\Delta t} = \frac{u_{(n+1)m} - 2u_{nm} + u_{(n-1)m}}{\Delta x^2} \iff$$

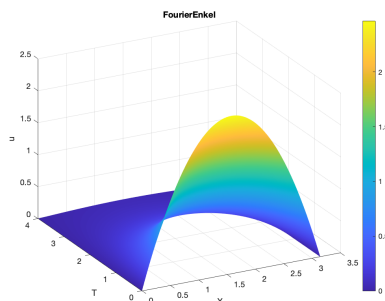
$$\iff u_{n(m+1)} = u_{nm} + \lambda(u_{(n+1)m} - 2u_{nm} + u_{(n-1)m}).$$

Vi har infört betäckningen  $\lambda = \Delta t / \Delta x^2$ . Detta låter oss implementera en algoritm där vi stegar från  $t = 0$  till vilken tidpunkt vi vill och beräknar funktionsvärdena rekursivt under tiden. Vi känner till att  $\lambda$  nödvändigtvis måste vara som störst  $1/2$  för att vi ska ha stabilitet. Vi bestämmer värdena för  $u$  vid  $t = 0$  utifrån begynnelsevillkoret  $u(x, 0) = g(x)$ . Vi vill avgöra noggrannheten

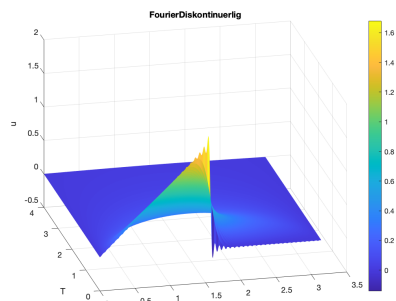
och ser att det numeriska fel vi gör har sitt ursprung från approximationen av derivatan. Därav bestäms felet i beräkningen utifrån hur vi väljer  $\Delta x$  och  $\Delta t$ . Specifikt vet vi utifrån noggrannhetsordningen av framåt differensen för  $u_t$  och centraldifferens av andra ordningen för  $u_{xx}$  att felet minskar enligt  $O(\Delta t + \Delta x^2)$ .

## Jämförelse

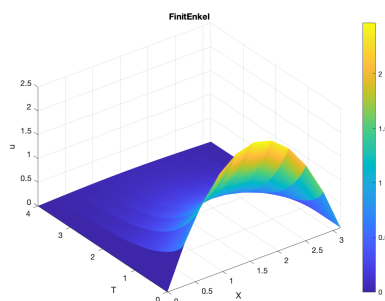
Vi har nu bekantat oss med hur teorin ser ut för detta problem. Genom implementation av dessa metoder i MATLAB kan vi plotta approximationer av  $u$ . Vi gör detta från  $t = 0$  till  $t = 4$ . Här kommer fyra olika plottar genererade av MATLAB:



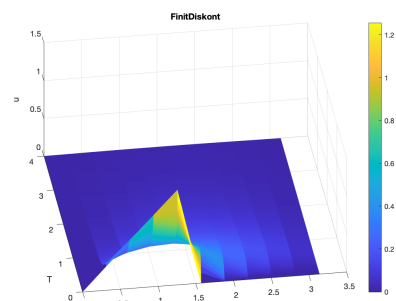
(a) Fourierlösning  $g(x) = x(\pi - x)$



(b) Fourierlösning intervallfunktion



(c) Finit differens  $g(x) = x(\pi - x)$



(d) Finit differens intervallfunktion

Figure 1: Vi undersöker prestation av finita differensmetoden och approximation med Fourierserier för olika begynnelsevillkor.

Vi jämför först lösningen med Fourierserier för de olika begynnelsevillkoren. För bild (a) används bara tio termer i Fourierserien, ändå är kurvan mycket

smidig. Däremot så används 100 termer i bild (b) eftersom svängningarna blev mycket märkbara över hela lösningen för färre termer. Det bekräftar vad vi redan hade sätt, nämligen att felet försvinner enligt betydligt snabbare för  $g(x) = x(\pi - x)$ , vilket är, medan vi angränsar vid divergens för det andra begynnelsevillkoret med feltermen bara en aning mindre än  $1/n$ . Bara utifrån detta kan vi säga att en Fourierlösning var mycket fördelaktig utifrån begynnelsevillkoret  $g(x) = x(\pi - x)$ . Sen har vi som sagt tidigare fortfarande problemet med Gibbs fenomen, att vi aldrig kommer få en lösning som konvergerar i just  $(\pi/2, 0)$ . Vi ser att även med 100 termer svänger lösningen vid denna diskontinuitet. Detta visar på att Fourierserier är svaga när det kommer till diskontinuerliga begynnelsevillkor. En slutsats blir att Fourierserier kan vara väldigt effektiva när det kommer till att hantera "snälla" begynnelsevillkor, eftersom vi inte behöver många termer för ett litet fel. De spricker dock med säkerhet för tuffare begynnelsevillkor, visserligen konvergerar medelkvadratfelet om vi väljer många termer, men vi får aldrig punktvis konvergens i diskontinuiteten (och konvergens runt omkring är långsam).

Med finita differensmetoden ser det lite annorlunda ut. Vi har som sagt påstått att felet växer som  $O(\Delta t + \Delta x^2)$  vilket vi noterar är oberoende av begynnelsevillkoret. Därmed är finita differensmetoden lite mer konsekvent än approximation med Fourierserier som varierade i noggrannhet. I de plottade bilderna delar vi bara upp  $x$ -ledet i tio steg för att vi ska se varifrån felet kommer. Vi förstår enkelt att felet kan göras godtyckligt litet om vi väljer en finare indelning, men felet bör minska enligt  $O(\Delta t + \Delta x^2)$ , alltså linjärt i  $\Delta t$ . Detta är märkbart sämre än i Fourierfallet där vi hade  $O(1/n^2)$  på  $t = 0$  och som vi kan se, exponentiellt avtagande med  $n$  om  $t$  är positivt. Om vi däremot är betydligt mer intresserade av att vi gör rätt vid begynnelsevillkoret så är alltid finita differensmetoden bättre eftersom vi faktiskt inte gör något numeriskt fel alls just där.

En jämförelse av Fourierserier och finita differensmetoden ger alltså att finita differensmetoden är mer konsekvent och bättre på att hantera ojämna begynnelsefunktioner, dock konvergerar den alltid en aning långsamt. Fourierserier har fördelen att de konvergerar betydligt snabbare för jämna ("snälla") begynnelsevillkor, men den får stora problem med ojämna begynnelsevillkor där den inte ens konvergerar i alla punkter och där konvergens i övrigt är mycket mer långsam.

## Konvergens av lösningarna

Vi vill också jämföra konvergens av metoderna i praktiken. Först med det kontinuerliga begynnelsevillkoret och lösningen med fourierserier kunde vi beskriva feltermen som en summa av termer på formen  $\sin(xn)/n^3$  ungefär, vilket vi noterade bör konvergera ungefär som  $O(1/n^2)$ . Notera att detta gäller när

$t = 0$  endast. Vid annat fall konvergerar felet exponentiellt. En summa av felen i olika punkter bör därmed begränsas av just  $\sin(xn)/n^3$ , termen som växer långsammast. Vi beräknar ett slags fel genom att lösa problemet med ett stort antal termer (2000) och beskriva felet 2-normen av matrisen motsvarande den finare lösningen och matrisen för ett färre antal termer. Vi plottar antalet termer mot felet och får följande:

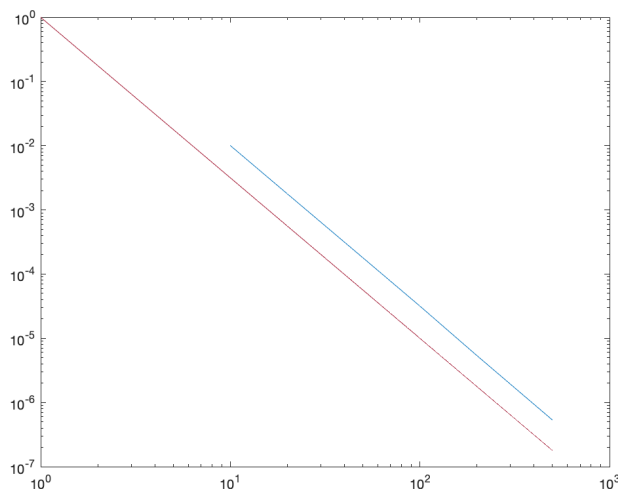


Figure 2: Blå linje motsvarar felet och röd linje är  $y = 1/n^{5/2}$

Vi ser istället att felet som funktion av antal termer minskar enligt  $O(1/n^{2.5})$ , vilket är bättre än förväntat. Tidigare har vi dock gjort överslagsberäkningar och sagt att felet minskar i värsta fall som  $O(1/n^2)$ . Det verkar som att den trigonometriska faktorn i täljaren förbättrar konvergensen med en faktor  $1/\sqrt{n}$ .

Vi kan på precis samma sätt plotta felet i fallet med diskontinuerligt begynnelsevillkor. Den dominerande termen var då ungefär  $\cos(nx)\sin(n)/n$ . Vi resonerade att nämnaren indikerar divergens  $O(1)$ , men att de oscillationerna från de trigonometriska funktionerna kan motverka nämnaren. Plot ger

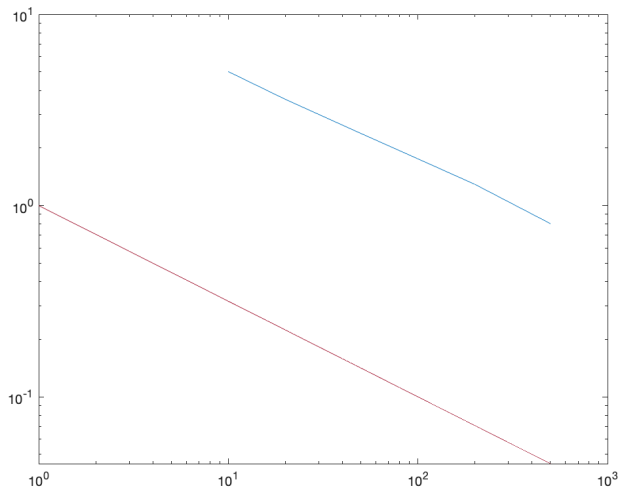
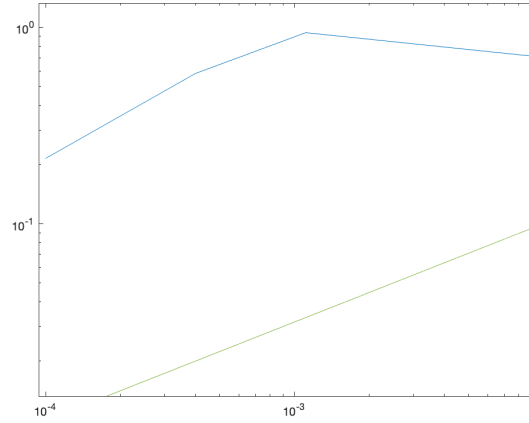


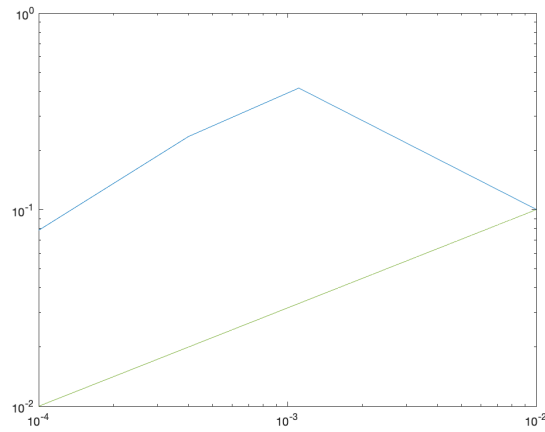
Figure 3: Blå linje motsvarar felet och röd linje är  $y = 1/\sqrt{n}$

Vi får att felet försvinner precis lika snabbt som  $O(1/\sqrt{n})$ , vilket indikerar igen att de trigonometriska funktionerna förbättrar konvergensen med en faktor  $1/\sqrt{n}$ . Konvergensen är som sagt tydligt långsammare än med det andra begynnelsevillkoret.

Vi kan försöka göra motsvarande för finita differensmetoden, men den är mer beräkningstung, vilket gör det svårt att mäta konvergensen för finare diskretiseringar. För båda begynnelsevillkor har vi beräknat felet i  $t = 1$ ,  $x = \pi/2$ . Felet definieras som en differensen mellan värdet vi får och värdet i en "fin" referenslösning. Vi plottar felet mot  $\Delta t$ . Resultatet är som följer:



(a)  $g(x) = x(\pi - x)$ . Felet i blått,  $y = \sqrt{x}$  i grönt.



(b) Intervallfunktion. Felet i blått,  $y = \sqrt{x}$  i grönt.

Vi ser ett mycket oberäkneligt beteende i för låga  $\Delta t$ , men senare ser lutningen ut att vara ungefär densamma som  $y = \sqrt{x}$ . Hur felet växer kan till stor del bero av valet av punkt. Hade vi exempelvis valt en punkt på randen hade vi som sagt inte fått något fel alls, för större värden  $t$  kan felen fortplantat på oväntade sätt. Vi ser att felet minskar med förfining av tidsstegen, men inte särskilt snabbt. Resultatet verkar indikera vad vi antog tidigare, nämligen att finita differensmetoden har liknande konvergens för olika typer av initialvillkor.



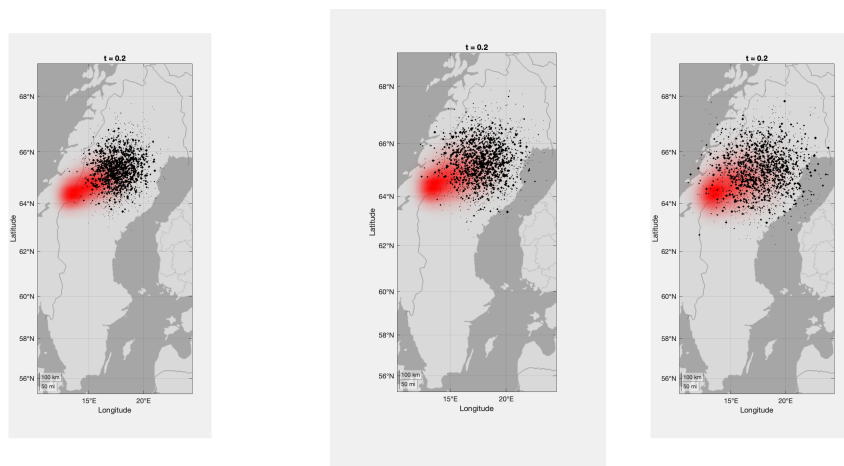
## Uppgift 4 Slumpvandring med Monte Carlo

Slutligen löser vi den lite mer komplexa differentialekvationen

$$\frac{\partial \rho}{\partial t}(y, t) + \operatorname{div}(v(y)\rho(y, t)) - c\Delta\rho(y, t) = 0$$

approximativt genom slumpvandring med Monte Carlo sampling. Vi genererar punkter och låter vinden i dessa ges av samma formel som i uppgift 2, nämligen  $v = (y, 1 - x)$ , sedan genererar vi initialvillkor med en tvådimensionell normalfördelning. Med Emanuel Ströms kod i canvas kan vi simulera positionen för olika punkter och deras förflyttning med tiden. Detta visar på ett ungefär hur alla rökpartiklar i verkligheten skulle förflytta sig.

Vi vill se hur förändring av diffusionskonstanten  $c$  påverkar lösningen. Rimligen kan vi anta en större spridning i alla riktningar, eftersom detta är i linje med vår intuitiva förståelse för diffusion. Här är resultat från några körningar:



(a) Lösning med  $c = 0.01$       (b) Lösning med  $c = 0.02$     (c) Lösning med  $c = 0.03$

Figure 5: Vi jämför partiklarnas spridning vid  $t = 0.2$  för olika diffusionskonstanter  $c$ .

Vi ser som väntat att partiklarna rör sig mer utspritt när diffusionskonstanten är större. Sen beror såklart lösningen också på saker som vinden och placering röken ursprungligen. Här är ett exempel där  $c = 0.01$  och vinden istället beskrivs enligt  $v = (1 - x, -y)$ . Då träffar en stor koncentration av röken olyckligtvis Stockholm.

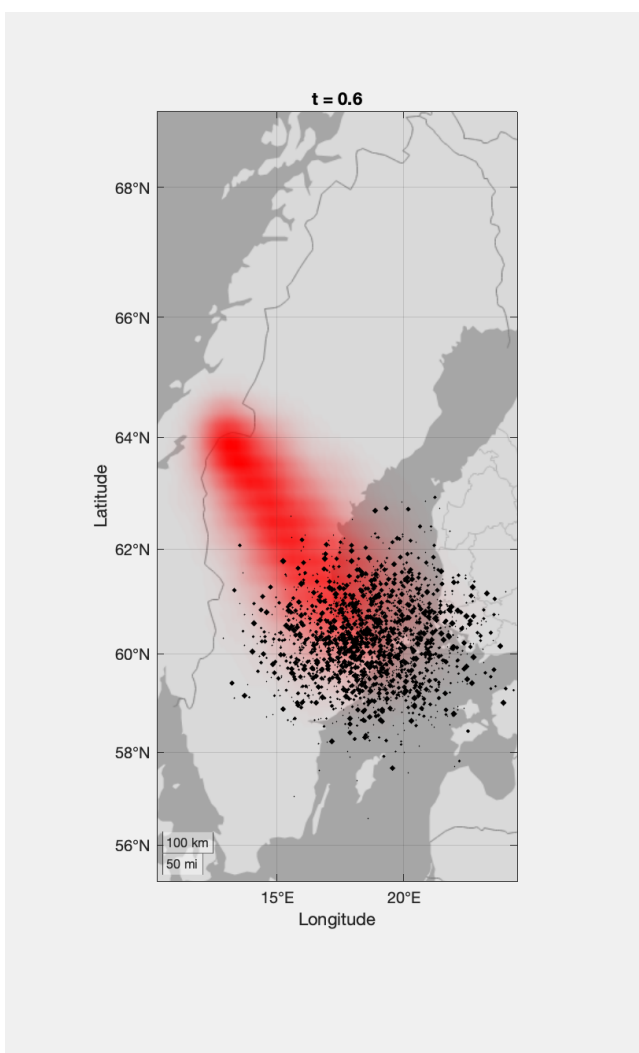


Figure 6: Bilden visar en lösning med vinden riktad höger ner i tidssteg  $t = 0.6$