Datorövning 1

Mål

Syftet med datorövningen är att du ska

- med hjälp av matlab kort repetera fourier-, sinus- och cosinus-serier och deras konvergensegenskaper,
- med hjälp av matlab experimentellt studera egenskaperna hos lösningar till endimensionella värmelednings- och svängningsproblem.
- se hur matlab kan användas för att visualisera lösningar.

Hjälpfiler som behövs för denna laboration kan hämtas från kursens hemsida, adress
www.maths.lth.se/~pellep/kurser/kontsys/
Matlab och maple kan hämtas från datordriftgruppen LTH, program.ddg.lth.se.

Förberedelser

- Läs igenom denna handledning samt avsnitt 3.1.2 och exempel 3.1 och 3.2 i läroboken. Titta också på anmärkningen på sidan 36 i läroboken.
- Fyll i svaren på frågorna på sidorna 11-12.
- Ta med läroboken och övningshäfte till övningen.

Fourierserier

En funktion f, som är given på intervallet $[0, L]$, kan ofta utvecklas antingen i trigonometrisk fourierserie

$$\frac{1}{2}a_0 + \sum_{k=1}^{\infty} a_k \cos(2\pi kx/L) + b_k \sin(2\pi kx/L)$$

eller i cosinusserie

$$\frac{1}{2}a_0 + \sum_{k=1}^{\infty} \alpha_k \cos(\pi kx/L)$$

eller i sinusserie

$$\sum_{k=1}^{\infty} \beta_k \sin(\pi kx/L),$$

där talföljderna $(a_k)_0^\infty$, $(b_k)_1^\infty$, $(\alpha_k)_1^\infty$ och $(\beta_k)_1^\infty$ kan beräknas då funktionen f är känd, se formelsamlingen. I denna laboration ska vi dels utgå från att funktionen är given av ett analytiskt uttryck, dels given grafiskt.

Funktioner givna av analytiska uttryck

Börja med att från kursens hemsida hämta filerna

```
delsum.m, filminsp.m, funklick.m, koeff.m, lab1pre.m, summera.m, swing.m,
varme.m
```

Starta matlab!
Med kommandot

```
lab1pre
```

Läser du in och definierar värdet på vissa variabler. Ta reda på vilka genom att skriva `who`. Notera att i denna laboration är `L = 1`. Ändra inte de nu satta värdena under resten av laborationen.
Här ska vi låta matlab räkna ut koefficienterna med hjälp av den numeriska metoden **snabb fouriertransformation (fast fourier transform)**, fft. Det finns naturligtvis ett matlabkommando som heter `fft`. Detta utnyttjas i `m`-filen `koeff` för att beräkna koefficienterna i den trigonometriska fourierserien och i halvperiodsutvecklingar med cosinus- och sinusserier. Valet mellan de tre alternativen sker genom att en fråga besvaras med `f`, `c` resp `s`. Det kommer också upp en matlabfigur där `akoeff` och `bkoeff` är plottade.
Genom att köra `lab1pre` får du bl a följande funktioner definierade.

```
funktion1 = 'max(sin(2*pi*X),0)';
funktion2 = '(X <L/2) - (X > L/2)' ;
funktion3 = 'X.*(X < L/4) + (L-X).*(X >= L/4)/3' ;
funktion4 = '(X<L/4).*X - (X>2*L/3).*(X<3*L/4)';
funktion5 = 'sin(10*pi*X).*(X > 0.4*L).*(X<0.5*L)';
```

Citationstecknen betyder att det som står innanför av matlab uppfattas som en sträng. Man ska tänka på det som en funktionsbeskrivning med hjälp av en regel, utan att värden beräknas. Här är `funktion1` en halvvågslikriktad sinusfunktion. Den bildning `(X > L/2)` som förekommer i `funktion2` är ett logiskt uttryck, som när man stoppar in en matris `X` skapar en matris av samma storlek som `X` där det står ettor på alla platser där matriselementet i `X` är $> L/2$ och nollor för övrigt. Uttrycket `(X > L/2)` beskriver alltså den förskjutna heavisidefunktionen $\theta_{L/2}$. Genom att tänka på sådana funktioner inser man att `funktion2` beskriver en fyrkantvåg, att `funktion3` beskriver en triangelvåg, och att `funktion4` och `funktion5` beskriver mer komplicerad funktioner.
Undersök nu några av dessa funktioner. De evalueras med hjälp av kommandot `eval`. Till exempel kommer

```
f=eval(funktion1);
```

att skapa en vektor som innehåller funktionsvärdena för `funktion1` i de punkter som står som komponenter i vektorn `X`. För att se hur funktionen ser ut, skriv

```
plot(X,f)
```

För laborationens fortsättning är det viktigt att denna funktionsvärdesvektor heter `f`. Det är sedan lätt att definiera om den med hjälp av `eval`-kommandot.
Kör nu `koeff` och `delsum` på de olika funktionerna, och prova de olika optionerna `f`, `c` och `s`. (Du kan avbryta `delsum.m` med `Ctrl-C`.) Studera koefficienternas uppträdande och seriernas konvergens i de olika fallen. Titta särskilt på överslängarna vid diskontinuitetspunkter eller i områden där funktionen växer kraftigt.

Funktioner givna genom sina grafer

Vi ska nu definiera funktioner genom att rita dem med hjälp av ett skript `funklick`. Efter det att kommandot skrivits kan man flytta runt ett hårkors i ett koordinatsystem med hjälp av musen, och lägga in ett antal punkter genom att klicka med vänster musknapp. *Låt första punkten ligga på y-axeln.* Det gör inget om x-koordinaten inte blir exakt 0, ty programmet justerar din ritning så att första punktens x-koordinat blir 0. Fortsätt att klicka på vänster musknapp där du vill ha en brytpunkt i den sträckvis linjära funktion, som ritningen producerar. Låt sista punkten få x-koordinaten 1. Programmet justerar så att sista punkten får exakt x-koordinaten 1. *Avsluta ritandet genom ett tryck på höger musknapp.* Om du har klickat på en punkt (x_1, y_1) och om du vill att nästa punkt ska vara (x_1, y_2) så ska du klicka på en punkt (x_2, y_2), där $x_2 < x_1$. Du erhåller då en funktion med språng då $x = x_1$. På detta sätt kan man lätt konstruera funktioner med språngdiskontinuiteter, vilka kan vara lämpliga testobjekt när man studerar fourierutvecklingar.
Rita ett par stycken funktioner, av varierande snällhetsgrad, och kör för var och en av dem `koeff` och `delsum` på samma sätt som i föregående uppgift. Utveckla en och samma funktion i fourier-, sinus- och cosinusserie, och konstatera hur olika graferna blir. Titta i de två senare fallen på de approximerande funktionernas halvperiods- och paritetsegenskaper. (Paritet = udda, jämn.) Det är intressant att studera hur delsummorna anpassar sig till funktionen när denna är komplicerad.

2 Värmeledningsekvationen och vågekvationen

Värmeledningsekvationen

Enligt Fouriers metod, eller variabelseparationsmetoden, gäller att ett värmeledningsproblem med dirichletvillkor,

$$\begin{cases} \dfrac{\partial u}{\partial t} - \dfrac{\partial^2 u}{\partial x^2} = 0, & 0 < x < L,\ t > 0, \\ u(0,t) = u(L,t) = 0, & t > 0, \\ u(x,0) = f(x), & 0 < x < L, \end{cases}$$

har en lösning av formen

$$u(x,t) = \sum_{1}^{\infty} b_k e^{-k^2\pi^2 t/L^2} \sin(k\pi x/L),$$

där $(b_k)_1^\infty$ är koefficienterna i sinusutvecklingen av begynnelsevärdesfunktionen $f(x)$. Lös nu värmeledningsproblemet för *några* olika val av begynnelsefunktioner f, dels ritade med hjälp av funklick, dels analytiskt givna av funktion1,...,funktion5. Ritkommandot funklick levererar en funktion kallad f. Om du använder till exempel den fördefinierade funktionen funktion1 behöver du tillverka funktionen f med kommandot f=eval(funktion1);. Beräkna koefficienterna b_k med hjälp av koeff. Vilken option f, c eller s ska användas? _______

Lösningsformeln ovan är inlagd i skriptet varme, som beräknar lösningen vid en från $t = 0$ växande följd av tidpunkter. Detta skript visar ej delsummorna av serien, utan summerar direkt 127 termer. Lägg särskilt märke till den kraftiga utjämningen hos lösningen med tiden, även från diskontinuerliga begynnelsedata.

Allteftersom som man stegar sig fram i tiden lagras lösningen radvis i en matris losn. Man kan sedan visa denna i en tredimensionell plot, genom att skriva mesh(losn). Använd även kommandot view för att ändra synvinkeln. Gör till exempel view(20,10), view(10,20) eller view(0,90), se help view. Du kan alternativt klicka på ikonen med en cirkel med en pil, som antyder rotation. Peka sedan på figuren med musen. Håll vänster musknapp nedtryckt samtidigt som du rör musen. Som du ser roteras boxen. När du släpper musen roteras figuren till det läge du flyttat boxen. Vinklarna visas i nedre vänstra hörnet då vänster musknapp hålls nedtryckt.

Värmeledningsekvationen ovan kan vara en modell för värmeledning i en stav med längden L. Ett mått på enegiinnehållet i staven vid begynnelsetiden $t = 0$ är

$$\int_0^L f(x)\,\mathrm{d}x.$$

Denna integral kan beräknas approximativt med en rektangelapproximation. Vad blir sum(f)/N? _________________

Undersök vad som händer med energin med växande tid, dvs beräkna $\int_0^L u(x,t)\,\mathrm{d}x$ för några olika tider t. Detta kan ske på följande sätt: Skriv sum(losn(j,:))/N för några heltalsvärden på j i intervallet $[1, 61]$. (Intervallets utseende beror på det sätt på vilket tiden beskrivs i varme.) Skriv ner värdena du fick! ___________________________

____

Vad händer med temperaturen $u(x,t)$ då t blir stort? _______________________

Lös även värmeledningsproblemet med neumannvillkor. Välj någon funktion som begynnelsevillkor. Vilka är randvillkoren i detta fall? ___________________________

Välj sedan en lämplig option i koeff. Vilken? _______

Hur ser lösningsformeln ut i detta fall? _______________________________

Kör varme. När varme är klart kan du köra mesh(losn) och betrakta lösningen grafiskt som ovan. Beräkna begynnelseenergin sum(f)/N: _________________

Beräkna också sum(losn(j,:))/N för några olika j i samma intervall som ovan.

Resultat: ____

Vad händer med temperaturen $u(x,t)$ då t blir stort? _______________________

Vågekvationen

På samma sätt visas med Fouriers metod att svängningsproblemet för en fast inspänd sträng,

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} - c^2 \dfrac{\partial^2 u}{\partial x^2} = 0, & 0 < x < L, t > 0, \\ u(0,t) = u(L,t) = 0, & t > 0, \\ u(x,0) = f(x),\ u'_t(x,0) = g(x), & 0 < x < L. \end{cases}$$

har lösningen

$$u(x,t) = \sum_1^\infty \big(b_k \cos(k\pi ct/L) + c_k \sin(k\pi ct/L)\big) \sin(k\pi x/L),$$

där $(b_k)_1^\infty$ är koefficienterna i sinusutvecklingen av begynnelselägesfunktionen f, och där $c_k k\pi c/L$ är koefficienterna i sinusutvecklingen av begynnelsehastigheten g.
Denna lösningsformel i fallet $c = 1$ och $g = 0$ finns programmerad i scriptet swing. Experimentera med några olika utseenden av begynnelselägesfunktionen f på samma sätt som i värmeledningsfallet. Låt till en början funktionerna vara kontinuerliga, och låt dem inte variera alltför kraftigt eller ha alltför många pucklar. Koefficienterna beräknas med hjälp av koeff, (option? _______) varefter man kör swing.
Ett bra sätt att åskådliggöra lösningen i detta fall är som en animation, där man i en film ser hur strängens utböjning varierar med tiden. Genom att köra skriptet filminsp lagrar matlab de successiva plottarna från swing i en matris M. Genom att sedan köra movie(M,antal) så visas filmen så många gånger som 'antal' anger. Man kan även påverka hastigheten i uppspelningen, se help movie. Undersök speciellt halvvågsfunktionen funktion1 och triangelvågen funktion3, jämför exempel 3.2 (gitarrsträngen) i läroboken.
Prova också med funktionerna f=sin(pi*X), f=sin(2*pi*X) och sin(3*pi*X) som begynnelseläge för en svängande sträng.
Vad kallas det fenomen man ser? ____________________________________

En funktion, som består av en smal puckel, är den via lab1pre.m definierade

```
funktion5 = 'sin(10*pi*X).*(X > 0.4*L).*(X<0.5*L)' ;
```

Använd denna som begynnelseläge och gör en film på samma sätt som ovan.
Beskriv i ord vad som händer. ____

____

3 Numeriska lösningar

Inledning

Matlab består, förutom grundpaketet, av ett antal verktygslådor (toolboxes), varav en är en PDE verktygslåda. Den handlar om numerisk lösning av partiella differentialekvationer med **finita elementmetoden**. I nästa avsnitt kommer vi att göra en mycket kort presentation av denna, se även kapitel 8 i läroboken. Finita elementmetoden är den ena av två förhärskande metoder för numerisk lösning av partiella differentialekvationer, där den andra är **finita differensmetoden**, se kapitel 2 i boken.
Kursen i stort går främst ut på att studera analytiska lösningsmetoder. Sådana finns bara i ganska speciella situationer, med enkla områden, differentialekvationer och randvillkor. De numeriska metoderna fungerar i mycket allmännare situationer, med 'godtyckliga' områden och där koefficienterna i differentialekvation och randvillkor får bero av rums- och tidsvariabler. Man finner dock ganska snart att för att effektivt och kritiskt kunna använda ett numeriskt programpaket, som matlabs PDE verktygslåda, så måste man vara väl förtrogen med den analytiska teorin.

Om finita elementmetoden

Antag att u är en lösning till randvärdesproblemet

$$\begin{cases} -\Delta u = f, & \text{i } \Omega, \\ \dfrac{\partial u}{\partial \boldsymbol{n}} + qu = g, & \text{på } \partial\Omega, \end{cases} \tag{1}$$

för ett område Ω med styckvis glatt rand $\partial\Omega$. Låt φ vara en C^1-funktion i en omgivning av Ω. Sådana funktioner kallar vi **testfunktioner**. Multiplicera differentialekvationen med φ och integrera över Ω. Med hjälp av en partialintegration, Greens formel I, får vi

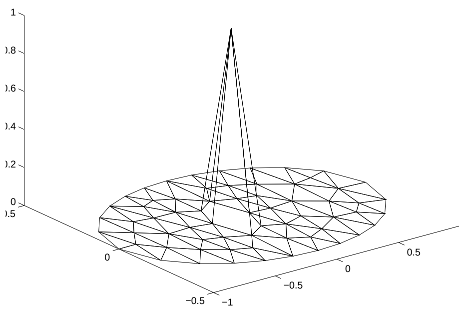
$$\begin{aligned} &\int_D (-\Delta u)\varphi \, dV = \int_D f\varphi \, dV \iff \\ &\int_D \operatorname{grad} u \cdot \operatorname{grad} \varphi \, dV - \int_{\partial D} \frac{\partial u}{\partial \boldsymbol{n}} \varphi \, dS = \int_D f\varphi \, dV. \end{aligned}$$

Insättning av randvillkoret ger

$$\int_D \operatorname{grad} u \cdot \operatorname{grad} \varphi \, dV + \int_{\partial D} qu\varphi \, dS = \int_{\partial D} g\varphi \, dS + \int_D f\varphi \, dV. \tag{2}$$

Om u är en lösning till (1) så gäller alltså (2) för varje testfunktion φ. Om (2) är uppfylld för alla testfunktioner sägs u vara en **svag lösning** till (1).
För att u ska vara en svag lösning till (1) krävs bara att u är en gång deriverbar, trots att andraderivator ingår i problemformuleringen (1). Situationen påminner om derivation i distributionsmening, fast där har man en annan klass av testfunktioner. Om Ω, f och q är tillräckligt reguljära kan man visa att varje svag lösning också är en "klassisk" lösning till (1).

Vid **finita elementmetoden** går man ett steg längre, och betraktar en *diskret* variant av (2). Börja med att göra en uppdelning av området Ω i trianglar, se figuren nedan, och ersätt klassen av testfunktioner ovan med klassen T av funktioner vars grafer är plana över var och en av trianglarna. Hörnen i trianglarna kallas **noder**. Ett viktigt exempel på en funktion i T är **tältfunktionen** τ_k i figuren nedan.



Den har värdet 1 i centralnoden (x_k, y_k) och 0 i alla andra noder. Det fina med tältfunktioner är att varje funktion $\varphi \in T$ kan skrivas

$$\varphi(x, y) = \sum_k \varphi(x_k, y_k)\, \tau_k(x, y).$$

Tänk efter att detta är riktigt. Tältfunktionerna utgör alltså en bas för T, och koefficienterna i utvecklingen av en funktion i denna bas är helt enkelt funktionens värde i centralnoden.

Vid en vanlig variant av finita elementmetoden, **Galerkins metod**, vill man att även lösningen u ska tillhöra klassen T. Detta leder till problemet:

Bestäm $u \in T$ så att (2) gäller för alla $\varphi \in T$.

Eftersom tältfunktionerna är en bas för T räcker det här att låta φ genomlöpa tältfunktionerna τ_j. För varje j är det lätt att räkna ut till exempel första integralen i (2),

$$\int_D \sum_k u(x_k, y_k)\, \mathrm{grad}\, \tau_k(x, y) \cdot \mathrm{grad}\, \tau_j(x, y)\, \mathrm{d}V.$$

Här är alla gradienter konstanta över varje triangel, och alla termer i summan försvinner utom det fåtal som svarar mot noder k som ingår i någon triangel tillsammans med noden j. På samma sätt hanteras de andra termerna i (2). Detta leder till ett kvadratiskt ekvationssystem

$$KU = F,$$

där U är en kolonnmatris som innehåller u:s värden i noderna, F är en kolonnmatris uträknad från f och g, och där koefficientmatrisen K kallas för **styvhetsmatrisen**. Beräkningen av K och F kallas **assemblering**.

På detta sätt övergår problemet att lösa randvärdesproblemet (1) till att lösa ett, i allmänhet mycket stort, linjärt ekvationssystem. Antalet obekanta är lika med antalet noder. Man kan visa att om Ω, f, q och g är tillräckligt reguljära, och om man väljer en svit av allt finare indelningar, så kommer de lösningar man får ur finita element-metoden att konvergera mot lösningen till (1). På liknande sätt kan man behandla allmännare differentialekvationer av Sturm-Liouvilletyp och allmännare randvillkor.

4 Matlabs PDE verktygslåda

Allmänt

Matlabs PDE verktygslåda kan användas för att lösa problem i två rumsdimensioner, till exempel randvärdesproblem för Laplace-Poissons ekvation, begynnelse-randvärdesproblem för värmelednings- och vågekvationen, samt egenvärdesproblem. Man kan göra väldigt mycket enbart med hjälp av menyer, men har man speciella krav finns även möjlighet att köra programmet med kommandon i matlab.

Problem

Vi kommer att se hur verktygslådan fungerar genom att studera några modellproblem. De är

$$\begin{cases} -\Delta u = 10, & \text{i } \Omega, \\ u = 0, & \text{på } \partial\Omega, \end{cases}$$

där $\Omega = \{(x,y); 0 < x < 1, 0 < y < 1\}$ är enhetskvadraten och med samma Ω

$$\begin{cases} -\Delta u = 0, & \text{i } \Omega, \\ u(x,0) = 0, u(x,1) = 10, & 0 < x < 1, \\ u(0,y) = 0, u'_x(1,y) = 0, & 0 < y < 1. \end{cases}$$

Sen ska samma problem studeras med ett lite mer komplicerat Ω som du får välja själv.

Användning av PDE verktygslådan

Starta toolboxen genom att ge kommandot pdetool. Det kommer upp ett fönster med ett antal "knappar" och en "ritbräda", där man ritar in sitt område. I fortsättningen kallar vi detta *pde-fönstret*. Man går ut ur toolboxen genom att välja Exit i File-menyn eller genom en knapp längst ner till höger i pde-fönstret.
När man öppnat pde-fönstret finner man överst en rad knappar med rubriker:

File, Edit, Options, Draw, Boundary, PDE, Mesh, Solve, Plot, Window, Help

Trycker man på dessa kommer det upp rullgardiner, som vi kommer att bekanta oss med efterhand.
I en andra rad i pde-fönstret finns en rad med diverse snabbknappar, för ofta förekommande kommandon. Vi kommer successivt att bekanta oss med dessa.
Vad knapparna i den översta raden ska användas till framgår av namnen: man anger typ och koefficienter i differentialekvation med Pde, området med Draw, randvärden med

Boundary, finita element-metodens triangelindelning med Mesh, beställer en lösning med Solve, och visualisering med Plot. Utförligare beskrivning följer nedan.
Kör man med menyer sker kontakten med data och lösningar enbart grafiskt. Man kan få ut sifferväden ur verktygslådan genom att välja alternativ export of solution bakom knappen Solve i översta knappraden i pde-fönstret.
När man är klar med en deluppgift och ska gå över till nästa gör man omstart genom att välj new under File-knappen.

Formulering och lösning av randvärdesproblem

Här ska vi bekanta oss med det grundläggande handhavandet av matlabs PDE verktygslåda, och experimentera med Poissons ekvation i ett lagom komplicerat område och ganska allmänna randvillkor. Följ nedanstående anvisningar.

- Områden kan ritas i pde-fönstret med hjälp av musen. För att underlätta detta är det bra att skaffa ett rutnät, genom att välja grid under Options-knappen. Välj sedan även snap under Options-knappen för att figurerna ska hamna rätt i förhållande till rutnätet.

- *Differentialekvation.* Välj pde specification under Pde-knappen. Man får då upp en dialogruta där man kan välja mellan elliptic, parabolic, hyperbolic eller eigenmodes. Vi behåller till en början förhandsinställningen, som är elliptic, med en Sturm-Liouvilleekvation

  $$-\operatorname{div}(c \operatorname{grad} u) + au = f,$$

  där c, a och f är valda så att man har Poissons ekvation $-\Delta u = 10$. Koefficientfunktionerna får bero av x och y, men till att börja med kan du behålla grundinställningen.

- *Generering av område.* Med hjälp av menyer kan man konstruera områden sammansatta av rektanglar, ellipser och polygoner. Dessa alternativ finns under Draw-knappen, men man kan även använda snabbknapparna med rektangel-, ellips- eller polygonsymboler. Ritandet går till så att man ställer markören någonstans, trycker på vänster musknapp och håller den nere samtidigt som man rör musen. När man släpper musknappen skapas en figur. (Polygonritandet fungerar lite annorlunda, men det märker du snabbt.) Symbolen + i mitten på en snabbknapp säger att figuren får sitt centrum i punkten där man startade.

  Börja till exempel med att rita en enhetskvadrat.

- *Randvillkor.* Välj boundary mode under Boundary-knappen eller snabbknappen $\partial\Omega$. Ränderna till de områden du ritat framträder då. Ta bort alla ovidkommande "inre" ränder genom att välja remove all subdomain borders under Boundary-knappen. Dubbelklicka sedan på något randstycke så kommer det upp en dialogbox. I denna väljes typ av randvillkor och koefficienter i dessa. Man har att välja mellan

$$\text{Dirichletvillkor } hu = r \text{ och Neumannvillkor } \boldsymbol{n} \cdot (c \operatorname{grad} u) + qu = g.$$

(Det senare är ett samlingsnamn på det vi kallat robinvillkor och neumannvillkor.) Koefficienterna får här vara funktioner av x och y. Grundinställningen är homogena dirichletvillkor, $u = 0$ på $\partial\Omega$. Krångla inte till det för mycket i början, utan ta dirichletvillkor $u = r$ med olika konstanta r på några randstycken, och homogena neumannvillkor $\boldsymbol{n} \cdot \operatorname{grad} u = 0$ på några andra randstycken. Randstycken med neumannvillkor färgas blå och randstycken med dirichletvillkor färgas röda i boundary mode. Vill man välja samma randvillkor längs hela randen behöver man inte klicka på de olika randstyckena utan väljer Specify Boundary Conditions under Boundary-knappen.

- *Triangelgenerering.* Generera en ”mesh” bestående av trianglar genom att välja initialize mesh under Mesh-knappen eller genom att trycka på snabbknappen med *en* triangel. Indelningen kan sedan förfinas genom att man väljer refine mesh under Mesh-knappen eller trycker på snabbknappen med *fyra* trianglar. Här gäller att man får högre noggrannhet i beräkningarna ju finare indelning man gör, men det tar naturligvis längre tid. Antalet noder och antalet trianglar står längst ner i pde-fönstret. *Varning:* Förfina inte indelningen alltför mycket, annars blir du aldrig klar.

- *Lös problemet.* Randvärdesproblemet löses genom att man väler solve pde under Solve-knappen eller trycker på snabbknappen =.

- *Visualisering.* Använder man förhandsinställningen visualiseras lösningen genom en färgkodad plot i pde-fönstret. Genom att välja parameters under Plot-knappen eller snabbknappen med en 3D-plot, får man upp en dialogruta där man kan påverka färkodningen, få nivåkurvor inlagda, få trianguleringen inlagd, etc. Genom att välja arrows i dialogrutan kan man få pilar som illustrerar gradienten. På detta sätt kan man till exempel visualisera värmeströmmar i stationära värmeledningsproblem. Genom att välja height (3D-plot) i dialogrutan får man en 3D-plot, som kan vridas och vändas genom att man ställer markören på figuren, trycker ner vänster musknapp, och flyttar markören.

Frågor att besvara: Hur stämmer lösningen med randvillkoren? Hur ser man i en 3D-plot av lösningen att den uppfyller homogena neumannvillkor? Vad betyder detta villkor i en 2D-plot med flödespilar, erhållna med arrows? Hur stämmer flödesbilden fysikaliskt med de källor och randvillkor du använt?

Upprepa proceduren för nästa problem och med nya lite mer komplicerade områden, ta till exempel ett L format område.

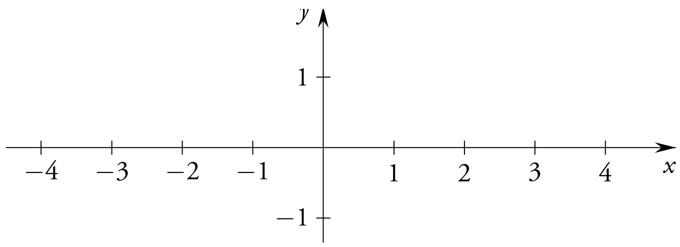
5 Förberedelsefrågor

1. Betrakta funktionen

$$f(x) = \begin{cases} x & \text{då } 0 \leqslant x \leqslant 1 \\ 1 & \text{då } 1 \leqslant x \leqslant 2 \end{cases}$$

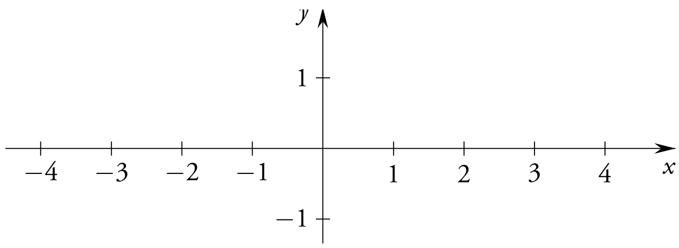
På intervallet $0 \leqslant x \leqslant 2$ kan funktionen f då beskrivas till exempel med en trigonometrisk fourierserie, en cosinusserie eller en sinusserie.

a) Rita i koordinatsystemet nedan den trigonometriska fourierseriens summa då $-4 \leqslant x \leqslant 4$.



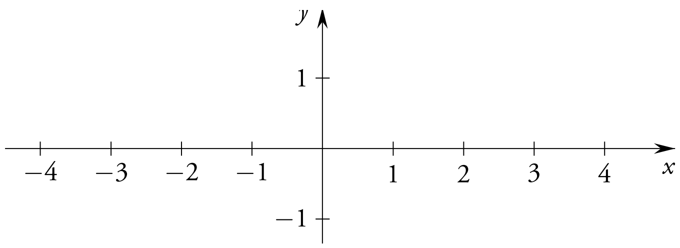
Vilken period T har den trigonometriska fourierserien? ____________________

b) Rita i koordinatsystemet nedan cosinusseriens summa då $-4 \leqslant x \leqslant 4$.



Vilken period T har cosinusserien? ____________________

c) Rita i koordinatsystemet nedan sinusseriens summa då $-4 \leqslant x \leqslant 4$.



Vilken period T har sinusserien? ____________________

2. Låt u vara lösningen till svängningsproblemet

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} - c^2 \dfrac{\partial^2 u}{\partial x^2} = 0, & 0 < x < L, t > 0, \\ u(0,t) = u(L,t) = 0, & t > 0, \\ u(x,0) = f(x), u'_t(x,0) = g(x), & 0 < x < L. \end{cases}$$

För att lösa problemet kan det vara lämpligt att utveckla u, f och g i lämpliga serier. Vilken typ av serie är lämplig? ____________________________

3. I figuren nedan är några nivåkurvor till en funktion u ritade. Rita i figuren in grad $u(A)$ och grad $u(B)$. Den exakta längden av gradienterna kan inte bestämmas men det ska framgå vilken gradient som är längst. Punkterna A och B finns i figuren.