Matlabövning 3

De redovisningar som skall göras skall vara inlämnade senast den 17:e oktober

Övningen redovisas antingen på papper eller per epost till

jea@math.chalmers.se

genom lösningen på deluppgifterna 2 och 4 . Deluppgifterna 1 och 3 används för att bygga upp lösningarna till nr 4 som skall redovisas. Man måste alltså gå igenom även dessa. Deluppgift 5 är helt frivillig och till för den som vill se rörliga bilder.

Matematiskt handlar övningen om Fouriers metod för att lösa vågekvationen (se avsnitt 20.7 i kursboken).

Vi väljer i den framställningen parametervärdena så att problemet blir att beskriva lösningarna till

$$u''_{xx} = u''_{tt}, \text{ för } 0 < x < \pi,\ t > 0$$

med randvillkoren $u(0,t) = u(\pi,t) = 0$ och begynnelsevillkoren $u(x,0) = f(x)$ och $u'_t(x,0) = 0$. Här bestämmer vi startformen $f(x)$ senare. Med dessa val kan lösningen $u(x,t)$ representeras som en serie

$$u(x,t) = \sum_{k=1}^{\infty} b_k \sin kx \cos kt$$

där koefficienterna är de man får då funktionen $f(x)$ utvecklas i en sinusserie på $[0,\pi]$, dvs

$$b_k = \frac{2}{\pi}\int_0^{\pi} f(x)\sin kx\, dx.$$

Vi skall se hur man kan använda MATLAB för att få en approximativ lösning

$$u_n(x,t) = \sum_{k=1}^{n} b_k \sin kx \cos kt.$$

1. För att integrera funktioner finns i MATLAB ett par metoder att välja på. Dels finns **quad** som bygger på Simpsons formel dels **quadl** som bygger på en annan metod. Normalt kräver den första flera beräknade funktionsvärden än den senare. Så låt oss i fortsättningen använda den senare. I grundformen ges integralen av en funktion fun på $[a,b]$ genom

   $>> quadl(fun, a, b)$

   men då uppstår frågan hur fun skall anges. Antag att vi vill integrera $\sin x$ på $[0,\pi]$. Vi kan göra det så här

   $>> quadl('sin(x)', 0, pi)$

   eller med $inline$ genom

   $>> f = inline('sin(x)'); quadl(f, 0, pi)$

   men det finns ytterligare ett sätt. Det finns möjlighet att hänvisa till en funktion genom ett så kallat **funktionshandtag** , vilket man kan göra genom att sätta @ framför funktionsnamnet. I fallet med sinusfunktionen kan vi därför få integralen genom

   $>> quadl(@sin, 0, pi)$

   Om du undersöker hjälpen till *quadl* ser du att det går att göra en del tillägg till det vi skrivit. Bland annat kan man styra precisionen men det går också att hantera parametervärden i integraler om dessa ges ett numeriskt värde.

Antag att vi istället vill integrera $\sin kx$ för parametervärdena $k = 1, 2, \ldots 10$. För att kunna hålla reda på vad som är variabel och vad som är parameter använder vi tillägg i *inline* så här

$>> fun = inline('sin(k * x)','x','k')$

Här skriver vi först *'x'* för att markera att den skall betraktas som variabel, de bokstäver som därefter räknas upp betraktas som parametrar. De integralvärden vi vill ha kan vi nu få i en vektor A genom

```
for k=1:10 A(k)=quadl(fun,0,pi,[],[],k);end
```

Vi måste här ha med de tomma [] på platserna 4 och 5 på grund av att dessa är reserverade speciella styrvärden (som vi inte bryr oss om här) men sedan kan vi ge värden på olika parametrar i den ordning de förekommer i *inline*, här bara *k*. Nu skall alla värdena finnas i vektorn A. **Ingen redovisning behövs**.

2. Vi skall här göra en fil **sinkoeff.m** och som anropas genom *B=sinkoeff(fun,n)* där *fun* är ett funktionshandtag eller en funktion definierad genom *inline* och positiva heltalet n angör hur långt i serieutvecklingen av funktionen vi vill gå. Har funktionen fouriersinkoefficienterna $b_1, b_2, \ldots$ ger

   B=sinkoeff(fun,n)

   B värdet $B = (b_1, \ldots, b_n)$.

   När vi skall skriva filen kommer vi att använda **feval** för att beräkna värdet av en funktion. Är funktionen given som ett handtag funger inte det vanliga $eval$.

   ```
   function B=sinkoeff(fun,n)
   % Funktionen skall anges som ett funktionshandtag.
   fsin=inline('feval(fun,x).*sin(k*x)','x','k','fun');
   for k=1:n
   B(k)=2*quadl(fsin,0,pi,[],[],k,fun)/pi;
   end
   ```

   Spara filen och testkör! **Redovisa genom att tala om steg för steg vad som sker i filen**.

3. Vi återvänder till vågekvationsproblemet och skapar en fil **vag.m** . Den skall anropas som *vag(x,t,fun,n)* och då ge summan

$$u_n(x,t) = \sum_{k=1}^{n} b_k \sin kx \cos kt$$

   där koefficienterna är de man får vid utveckling av *fun* i en sinusserie på $[0, \pi]$. En sådan fil kan se ut så här.

   ```
   function vag=vag(x,t,fun,n)
   B=sinkoeff(fun,n)
   vag=0;
   for k=1:n
   vag=vag+B(k)*sin(k*x)*cos(k*t);
   end
   ```

4. Här skall vi använda *vag(x,t,fun,n)* för att få fram en approximativ lösning $u_n(x,t)$ till det ursprungliga systemet då vi som funktionen $f(x)$ har

$$f(x) = 2(\pi/2 - |x - \pi/2|)/\pi.$$

Välj först ett n-värde som gör att $u_n(x,0)$ ser ut att vara en hyfsad approximation av $f(x)$, genom jämföra dem i en graf. Rita sedan ut ytan $z = u_n(x,t)$ över en rektangel $0 \leq x \leq \pi$, $0 \leq t \leq t_0$ där du väljer t_0 själv men som du tycker ger en bra bild av situationen.

Redovisa figuren!

5. Nu kan vi avsluta filtillverkandet med en fil som ritar ut den approximation av $u(x,t)$ som vi får genom *vag(x,t,fun,n)* för fixt t. Vi kallar denna fil **ritavag.m** och den kan ser ut så här.

```
function ritavag(t,fun,n)
x=[0:0.01:1]*pi
plot(x,vag(x,t,fun,n));
```

Pröva denna för några t- och n-värden då vi som funktion har

$$f(x) = 2(\pi/2 - |x - \pi/2|)/\pi.$$

Testa sedan vad som händer om du för ditt n-värde från förra deluppgiften skriver

```
for k=0:31
ritavag(k*pi/16,fun,n)
axis([0 pi -1.5 1.5])
M(k+1)=getframe
end
```

och när detta är utfört gör

$>> movie(M,3)$

Ingen redovisning