Matlabövning 3

De redovisningar som skall göras skall vara inlämnade senast den 14:e oktober

Övningen redovisas på papper genom lösningarna på deluppgifterna 1,3 och 4. Deluppgiften 2 används för att bygga upp lösningen till nr 3 som skall redovisas. Man måste alltså gå igenom även denna.

Bakgrund

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 genom att ersätta serien med en ändlig delsumma

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

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 f på $[a,b]$ genom

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

men då uppstår frågan hur f 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 variabeln, 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*. Därefter finns alla värdena i vektorn A.

Uppgifterna

1. 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 anger hur långt i serieutvecklingen av funktionen vi vill gå. Har funktionen fouriersinkoefficienterna $b_1, b_2, \ldots$ skall

   B=sinkoeff(fun,n)

   ge B värdet $(b_1, \ldots, b_n)$.

   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 eller genom inline.
   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**.

2. 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
   ```

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

$$f(x) = 1 - |1 - 2x/\pi|.$$

Välj först ett n-värde som gör att du tycker $u_n(x,0)$ ser ut att vara en hyfsad approximation av $f(x)$, genom jämföra dem i en graf. Observera att man i Matlab får absolutbelopp genom funktionen *abs*.

Redovisa figuren och ange vilket n-värde ni väljer!

4. Nu skall vi avsluta filtillverkandet med en fil som, om man bestämmer ett fixt *t*, ritar ut på intervallet $0 \leq x \leq \pi$ den approximation av $u(x,t)$ som vi får genom *vag(x,t,f,n)* för det fixerade t-värdet. Filen skall heta **ritavag.m** och den skall anropas så här: *ritavag(t,f,n)*.

Använd sedan denna för att med samma funktion som i föregående uppgift dvs

$$f(x) = 1 - |1 - 2x/\pi|$$

och det n du bestämde i föregående uppgift rita *vag(x,t,f,n)* för $t = 0, 1, 2, 3$ i en gemensam figur.

Redovisa filen och figuren.

5. Här kommer en rörlig illustration som inte behöver redovisas. Testa vad som händer om du för ditt n-värde och funktion f från förra deluppgiften skriver

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

och när detta är utfört gör

$>>$movie(M,5)