Kunnskap for en bedre verden

Institutt for matematiske fag

Eksamensoppgave i **TMA4125 BARE TULL - LF**

Faglig kontakt under eksamen:

Tlf:

Eksamensdato: 8.april-5. juni 2019

Eksamenstid (fra–til): 00:00 - 24:00

Hjelpemiddelkode/Tillatte hjelpemidler: Kode C: Bestemt, enkel kalkulator, og vedlagte formelark.

Annen informasjon:
Denne eksamenen består av 10 delpunkt som alle teller like mye. Alle svar skal begrunnes. Valgfritt programmeringsspråk i programmeringsoppgaver. Lykke til.

Målform/språk: bokmål

Antall sider: 8

Antall sider vedlegg: 0

Kontrollert av:

Informasjon om trykking av eksamensoppgave	
Originalen er:	
1-sidig ☐	**2-sidig** ☒
sort/hvit ☒	**farger** ☐
skal ha flervalgskjema ☐	

Dato Sign

Oppgave 1 Laplacetransformen til en funksjon er gitt ved

$$\mathcal{L}(f) = \int_0^\infty f(t)e^{-st}\, dt = F(s).$$

a) Dersom f er en integrerbar funksjon, kan vi skrive

$$|\int_0^\infty f(t)e^{at}\, dt| \le \int_0^\infty |f(t)e^{-st}|\, dt$$

og siden e^{-st} alltid er større enn 0, kan vi skrive

$$\int_0^\infty |f(t)e^{at}|\, dt = \int_0^\infty |f(t)|e^{-st}\, dt.$$

Dersom

$$|f(t)| \le Me^{at}$$

for to konstanter M og a, kan vi si at

$$\int_0^\infty f(t)|e^{-st}|\, dt \le \int_0^\infty Me^{at}e^{-st}\, dt = M\int_0^\infty e^{(a-s)t}\, dt,$$

og dersom $s > a$, slik at $s - a$ er negativ, konvergerer dette integralet. Vi oppsummerer alle ulikhetene og beregner det siste integralet:

$$|\int_0^\infty f(t)e^{at}\, dt| \le \int_0^\infty |f(t)|e^{-st}\, dt \le M\int_0^\infty e^{(a-s)t}\, dt = \frac{M}{s-a}.$$

b) Det er klart at $f(t) = e^{at}$ er en integrerbar funksjon som tilfredsstiller

$$|e^{at}| \le Me^{bt}$$

for to konstanter b og M. Man kan for eksempel velge $b = a$ og $M = 1$. La $s > a$. Vi beregner:

$$\int_0^\infty f(t)e^{at}\, dt = \int_0^\infty e^{at}e^{-st}\, dt = \int_0^\infty e^{(a-s)t}\, dt = \frac{1}{s-a}.$$

Oppgave 2 Siden

$$\int_{-\infty}^\infty f^2(x)\, dx = \int_{-1}^1\, dx = 2$$

er det klart at det er trygt å fouriertransformene f. Vi beregner

$$\frac{1}{\sqrt{2\pi}}\int_{-\infty}^\infty f(x)e^{-iwx}\, dx = \frac{1}{\sqrt{2\pi}}\int_{-1}^1 e^{-iwx}\, dx = \frac{1}{\sqrt{2\pi}}\frac{-1}{iw}\left(e^{-iw} - e^{iw}\right) = \sqrt{\frac{2}{\pi}}\frac{\sin w}{w}.$$

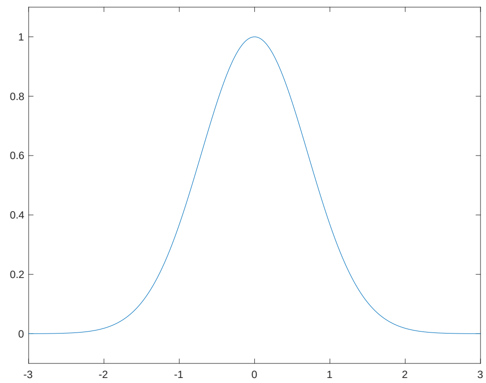
Oppgave 3 Her bruker vi D'Alemberts formel

$$u(x,t) = \frac{1}{2}\left(f(x+ct) + f(x-ct)\right) + \frac{1}{2c}\int_{x-ct}^{x+ct} g(v)\,dv$$

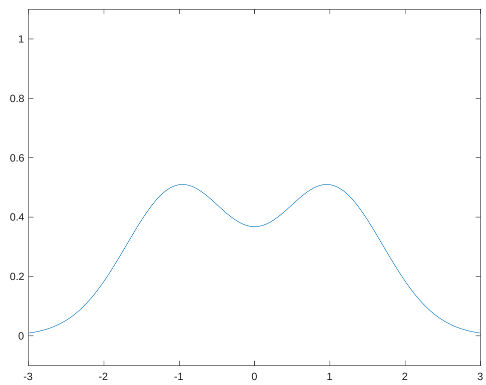
med $f(x) = e^{-x^2}$ og $g(x) = 0$, og $c = 1$. Vi får

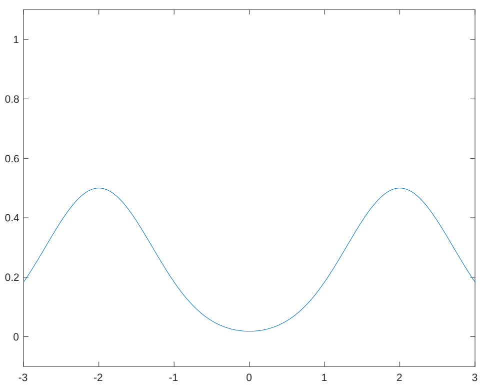
$$u(x,t) = \frac{1}{2}\left(e^{-(x+t)^2} + e^{-(x-t)^2}\right)$$

Her er et plot av funksjonen $u(x,0) = e^{-x^2}$:



Funksjonen $e^{-(x-t)^2}$ ser helt lik ut, men har maksimum i $x = t$ istedet for $x = 0$. Her er et plot av $u(x,1) = \frac{1}{2}\left(e^{-(x+1)^2} + e^{-(x-1)^2}\right)$:





For pedagogikkens skyld tar vi med et plot av funksjonen $u(x,1) = \frac{1}{2}\left(e^{-(x+2)^2} + e^{-(x-2)^2}\right)$: Merk likheten mellom $u(x,t)$ og bølgene som oppstår etter du har kastet en stein ned i et endimensjonalt vannspeil.

Oppgave 4 Det er klart at vi skal løse likningen

$$x^2 = \sin x$$

og at vi må skrive om til formen

$$x = g(x)$$

og kjøre fikspunktmetoden

$$x_{n+1} = g(x_n).$$

Omskrivningen til $x = g(x)$ kan gjøres på to åpenbare måter, nemlig $x = \sqrt{\sin x}$ eller $x = \arcsin x^2$. Vi tar en liten analyse. (Dette er ikke spurt etter på eksamen, men vi skal jo være litt pedagogiske.) Dersom $g(x) = \sqrt{\sin x}$, er

$$g'(x) = \frac{\cos x}{2\sqrt{\sin x}}.$$

På intervallet $[0.5, 1]$ har $\cos x$ maksimum i $x = 0.5$, og $\sqrt{\sin x}$ minimum i $x = 0.5$, slik at g' får et maksimum i $x = 0.5$. Vi bruker kalkulator, og beregner

$$g'(0.5) = \frac{\cos 0.5}{2\sqrt{\sin 0.5}} \approx 0.633719990281450$$

ser vi at $g'(x) < 1$ på intervallet $[0.5, 1]$. Siden $g(0.5) = \sqrt{\sin 0.5} \approx 0.692405617109078 > 0.5$ og $g(1) = \sqrt{\sin 1} \approx 0.917317275978108 < 1$ og g er monoton, ser vi at $0.5 < g(x) < 1$, og vi kan derfor garantere at fikspunktmetoden konvergerer til løsningen dersom $x_0 \in [0.5, 1]$. Vi prøver $x_0 = 0.7$, og får

x_0	0.802631725785675
x_1	0.848049023110869
x_2	0.866020417940237
x_3	0.872786773090119
x_4	0.875284497773173
x_5	0.876199653817684
x_{10}	0.876722810907523

Etter rundt førti iterasjoner ser metoden ut til å konvergere til $x \approx 0.876726215395062$. Den andre omskrivningen $x = \arcsin x^2$ konvergerer til $x = 0$. Dette kan man også forutse ved å analysere

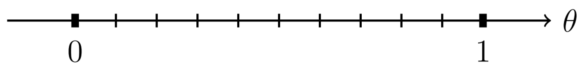
$$g'(x) = \frac{2x}{\sqrt{1 - x^4}}.$$

Siden $g'(.5) \approx 1.032795558988644$ og g er monotont stigende på $[0.5, 1]$, vil ikke fikspunktmetoden konvergere mot roten. Hvis du prøver med $x_0 = 0.7$, vil du se at iterasjonen konvergerer mot $x = 0$.

Oppgave 5 Chebyshevs nullpunktgitter med $n + 1$ punkter får man ved formelen

$$x_k = \cos \pi \frac{2k + 1}{2n + 2} \qquad 0 \le k \le n.$$

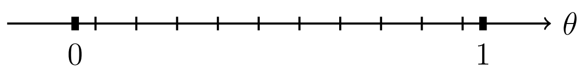
Tenk at du tar et ekvidistant gitter på $[0, \pi]$ med $n + 2$ punkter, der endepunktene er inkludert:



Dette gitteret er gitt ved formelen

$$\theta_k = \frac{\pi k}{n + 1} \qquad 0 \le k \le n + 1.$$

Nå flytter vi alle punkter et halvt knepp til den ene siden, og tar vekk et punkt:



Dette gitteret er gitt ved formelen

$$\theta_k = \frac{\pi(2k+1)}{2n+2} \qquad 0 \le k \le n.$$

Merk at nå er endepunktene ikke med. Hvis vi cosinustransformerer dette gitteret, får vi Chebyshevs nullpunktgitter, gitt ved den første formelen. Vi bruker $n = 3$ for å få fire punkter. Dette gir punktene

x_0	-0.923879532511287
x_1	-0.382683432365090
x_2	0.382683432365090
x_3	0.923879532511287

For å flytte gitteret fra $[-1, 1]$ til $[0, 1]$, kan vi bruke formelen

$$y_k = \frac{x_k + 1}{2}.$$

slik at gitteret blir

x_0	0.03806023374435
x_1	0.308658283817455
x_2	0.691341716182545
x_3	0.961939766255643

Lagranges interpolasjonspolynom på dette gitteret er:

$$\begin{aligned} p(x) = e^{x_0} \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} + e^{x_1} \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} \\ + e^{x_2} \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} + e^{x_3} \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}. \end{aligned}$$

For Chebyshevs nullpunkttgitter finnes et pent feilestimat:

$$|f(x) - p(x)| \le \frac{(b-a)^{n+1}}{2^{2n+1}} \max_{s \in [a,b]} |f^{n+1}(s)|$$

Her er $b - a = 1$, og $n = 3$. $f''''(x) = e^x$, som har maksimum i $x = 1$. Vi kan derfor konkludere at

$$|e^x - p(x)| \le \frac{e}{2^7} \approx 0.021236576784836.$$

Oppgave 6 Dette kan gjøres på forskjellige måter. Det enkleste er å observere at dette er bare Simpsons metode på intervallet $[0, 2]$. Simpsons metode er en Guass-Lobatto-regel med $n = 3$, og siden intervallene $[-1, 1]$ og $[0, 2]$ er like lange, kan vi plukke vektene fra tabellen, og få $A_0 = A_2 = \frac{1}{3}$ og $A_1 = \frac{4}{3}$. Kommer man ikke på dette er Gauss-Lobatto med $n = 3$, kan man beregne vektene ved formelen

$$A_k = \int_0^2 l_k(x)\, dx$$

der l_k er lagrangepolynomet tilhørende punkt k. Disse må da settes opp, ganges ut, og integreres.

Oppgave 7 Eulers implisitte metode blir

$$y_{k+1} = y_k - 2hy_{k+1}$$

eller

$$y_{k+1} = \frac{y_k}{1 + 2h}.$$

Matlab:

```
n=10;
h=1/n;
y=zeros(n+1,1);
y(1)=1;

for i=1:n
    y(i+1)=y(i)/(1+2*h);
end
disp(y(n+1))
```

Python:

```
import numpy as np

n=10.0;
h=1/n;
y=np.zeros(int(n+1));
y[0]=1;

for i in range(int(n)):
    y[i+1]=y[i]/(1+2*h)

print y[int(n)]
```

Kjører man kodene, får man $y_{10}0.16150558289$. Dette går fint å gjøre på kalkulator.

Oppgave 8

a) Vi gitrer opp x-aksen med gitteravstand h, og t-aksen med gitteravstand k. Et typisk gitterpunkt blir $(x_i, t_j) = (ih, jk)$, og vi skriver

$$u_{ij} \approx u(x_i, t_j).$$

Setter vi inn approksimasjonene

$$u_t(x_i, t_j) \approx \frac{u_{i,j+1} - u_{ij}}{k}$$

og

$$u_{xx}(x_i, t_j) \approx \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h^2},$$

inn i varmelikningen, får vi det eksplisitte skjemaet:

$$u_{i,j+1} = u_{ij} + \frac{k}{h^2}\left(u_{i+1,j} - 2u_{ij} + u_{i-1,j}\right).$$

b) Matlab:

```
n=5;    h=1/n;
m=100;  k=1/m;
r=k/h^2;

x=0:h:1;

u=zeros(n+1,m+1);
u(:,1)=x.*(1-x);

for j=1:m
   for i=2:n
       u(i,j+1)=u(i,j)+r*(u(i+1,j)-2*u(i,j)+u(i-1,j));
   end
end
```

Python:

```
import numpy as np

n=5.0
h=1/n
m=100.0
k=1/m
r=k/h**2

x=np.linspace(0,1,int(n)+1)

u=np.zeros((int(n)+1,int(m)+1))
u[np.arange(int(n+1)),0]=x*(1-x)

for j in range(1,int(m)):
    for i in range(2,int(n)+1):
        u[i-1,j]=u[i-1,j-1]+r*(u[i,j-1]-2*u[i-1,j-1]+u[i-2,j-1])
```