**Norges teknisk–
naturvitenskapelige universitet
Institutt for matematiske fag**

Faglig kontakt under eksamen:
Einar Rønquist, tlf. 73 59 35 47

EKSAMEN I FAG SIF5040 NUMERISKE METODER
Tirsdag 15. mai 2001
Tid: 09:00–14:00

Hjelpemidler: C1 – Alle trykte og håndskrevne hjelpemidler tillatt.
Alle kalkulatortyper tillatt.

Sensuren faller i uke 23.

Besvarelsen skal inneholde så mange mellomregninger at det tydelig går fram hvilke metoder og mellomresultater som er anvendt.

Oppgave 1 I denne oppgaven skal vi se på polynom-interpolasjon og spline-interpolasjon av funksjonen

$$f(x) = (e^x - 1)(3 - x), \quad 0 \le x \le 3.$$

Vi skal interpolere funksjonen i punktene $x_i = 0, 1, 2, 3$, d.v.s., vi har datasettet:

i	0	1	2	3
x_i	0	1	2	3
f_i	0	3.43	6.39	0

a) Finn interpolasjonspolynomet $p(x)$ av lavest mulig grad som interpolerer funksjonen $f(x)$ i punktene $x_i = 0, 1, 2, 3$.

Fasit: Lagrange interpolasjonsformel gir

$$\begin{aligned} p(x) &= 3.43 \cdot \frac{x(x-2)(x-3)}{1(1-2)(1-3)} + 6.39 \cdot \frac{x(x-1)(x-3)}{2(2-1)(2-3)} \\ &= 1.715x(x-2)(x-3) - 6.39x(x-1)(x-3). \end{aligned}$$

Hvis man vil skrive ut polynomet, da blir det :

$$\begin{aligned} p(x) &= x(x-3)\Big[1.715(x-2) - 6.39(x-1)\Big] \\ &= -1.4762x^3 + 4.1867x^2 + 0.7261x. \end{aligned}$$

Eller ved Newtons dividerte differenser:

x_k	f_x	f[,]	f[,,]	f[,,,]
0	0			
		3.43		
1	3.43		-0.24	
		2.95		-1.47
2	6.38		-4.665	
		-6.38		
3	0			

d.v.s.,

$$p(x) = 3.43x - 0.24x(x-1) - 1.47x(x-1)(x-2).$$

Dette er egentlig det samme polynomet som man finner ved Lagrange fromlen.

b) Finn den naturlige kubiske spline $S(x)$ som interpolerer funksjonen $f(x)$ i punktene $x_i = 0, 1, 2, 3$. (Du begynner med å beregne $z_i = S''(x_i)$. Splines funksjonen $S(x)$ kan skrives ved utrykk av z_i.)

Hva blir $S(2.5)$?

Fasit: Splines $S(x)$ defineres slikt:

$$S(x) = S_i(x), \quad \text{for } x_i < x < x_{i+1}, \quad i = 0, 1, 2.$$

La $z_i = S''(x_i)$, $i = 0, 1, 2, 3$. For naturlig kubisk splines, har vi $z_0 = 0$ og $z_3 = 0$, derfor trenger vi bare å beregne z_1, z_2. Ifølge algoritmen i læreboka Cheney og Kincaid s.334-335, trenger vi å løse et lineært lingningssystem:

$$h_{i-1}z_{i-1} + 2(h_i + h_{i-1})z_i + h_i z_{i+1} = 6(b_i - b_{i-1}), \quad i = 1, 2,$$

hvor

$$h_i = x_i - x_{i-1} = 1, \qquad b_i = (f_{i+1} - f_i)/h_i = f_{i+1} - f_i.$$

Skriv ut systemet:

$$\begin{aligned} 4z_1 + z_2 &= 6(f_2 - 2f_1 + f_0) = -2.88 \\ z_1 + 4z_2 &= 6(f_3 - 2f_2 + f_1) = -55.98 \end{aligned}$$

Løsningen er

$$z_1 = 2.964, \quad z_2 = -14.736.$$

Formel (5) i læreboka s.334 gir

$$S_i(x) = \frac{z_{i+1}}{6h_i}(x - x_i)^3 - \frac{z_i}{6h_i}(x - x_{i+1})^3 + \left(\frac{f_{i+1}}{h_i} - \frac{z_{i+1}h_i}{6}\right)(x - x_i) - \left(\frac{f_i}{h_i} - \frac{z_i h_i}{6}\right)(x - x_{i+1}).$$

Sett inn $h_i = 1$, $x_i = i$, vi får:

$$S_i(x) = \frac{z_{i+1}}{6}(x - i)^3 - \frac{z_i}{6}(x - i - 1)^3 + \left(f_{i+1} - \frac{z_{i+1}}{6}\right)(x - i) - \left(f_i - \frac{z_i}{6}\right)(x - i - 1).$$

For $x = 2.5$, ligger punktet mellon x_2 og x_3, derfor må S_2 brukes. Vi får:

$$S(2.5) = S_2(2.5) = 4.111.$$

Oppgave 2 Strømmen i et krets er gitt som en funksjon av tiden

$$I(t) = A\,(\sin(t))^B\, e^{Ct}, \tag{1}$$

hvor A, B, C er konstante parametre. I et eksperiment målte vi følgende data:

t	0.2	0.4	0.6	0.8	1.0
$I(t)$	0.35	0.55	0.53	0.40	0.35

Konstantene A, B, C bestemmes slik at funksjonen (1) tilpasser dataene best mulig. Omskriv funksjonen (1), beregn deretter konstantene A, B, C ved bruk av lineær minste kvadraters metode.

Hva blir strømmen når $t \to \infty$?

Fasit: Ta logaritmen på begge sider av ligningen:

$$\ln(I) = \ln A + B\ln(\sin t) + Ct$$

La $a_0 = \ln A$, $a_1 = B$, og $a_2 = C$. Normalligningene blir

$$\begin{aligned}
5a_0 + a_1\sum_{k=1}^{5}\ln(\sin t_k) + a_2\sum_{k=1}^{5} t_k &= \sum_{k=1}^{5}\ln I_k \\
a_0\sum_{k=1}^{5}\ln(\sin t_k) + a_1\sum_{k=1}^{5}(\ln(\sin t_k))^2 + a_2\sum_{k=1}^{5} t_k\ln(\sin t_k) &= \sum_{k=1}^{5}\ln(\sin t_k)\ln I_k \\
a_0\sum_{k=1}^{5} t_k + a_1\sum_{k=1}^{5} t_k\ln(\sin t_k) + a_2\sum(t_k)^2 &= \sum_{k=1}^{5} t_k\ln I_k.
\end{aligned}$$

Sett inn verdiene av t_k og I_k,

$$\begin{aligned}
5a_0 + -3.6356a_1 + 3.0000a_2 &= -4.2487 \\
-3.6356a_0 + 3.9681a_1 + -1.4818a_2 &= 3.1089 \\
3.0000a_0 + -1.4818a_1 + 2.2000a_2 &= -2.6129
\end{aligned}$$

som gir løsningen

$$a_0 = 1.5494, \quad a_1 = 1.2967, \quad a_2 = -2.4272$$

d.v.s.

$$A = 4.7086, \quad B = 1.2967, \quad C = -2.4272.$$

Siden C er negativ, går strømmen mot 0 når $t \to \infty$.

Oppgave 3 Vi skal finne null-punkter til funksjonen $f(x) = \sin(x)$ ved numeriske metoder. Vi ser at $x_1 = 0$ og $x_2 = \pi$ er begge eksakt løsningen til $f(x) = 0$.

a) Først, skal vi løse problemmet ved fikspunkt iterasjon. La

$$x = g_F(x) \quad \text{hvor} \quad g_F(x) = x + \sin x.$$

Sett opp iterasjons-skjema.

La startverdi $x^0 = 3.0$ (som ligger nær løsningen $x_2 = \pi$) og kjør 2 iterasjoner. La nå startverdi $x^0 = 0.1$ (som ligger nær løsningen $x_1 = 0$) og kjør 6 iterasjoner. Hva får du? Forklar resultatet.

Fasit: Iterasjons-skjemaet blir:

$$x^{k+1} = g_F(x^k) = x^k + \sin x^k, \quad k = 0, 1, 2, \cdots.$$

For $x^0 = 3.0$ gir 2 iterasjoner

$$x^1 = 3.14112000805987, \quad x^2 = 3.14159265357220.$$

For $x^0 = 0.1$ gir 6 iterasjoner

$$x^1 = 0.19983341664683, \quad x^2 = 0.39833948190924,$$

$$x^3 = 0.78622784960208, \quad x^4 = 1.49392106408408,$$

$$x^5 = 2.49096761603097, \quad x^6 = 3.09665148579400,$$

Vi ser at vi finner alltid løsningen $x_2 = \pi$, selvom vi velger startverdi veldig nær $x_1 = 0$. Dette er fordi $|g_f'(x)| > 1$ nær x_1, og derfor fikspunkt iterasjon konvergerer aldri mot x_1. Men $|g_f'(x)| < 1$ nær x_2, og derfor fikspunkt iterasjon konvergerer mot x_2.

b) Vi bruker nå Newtons metode. Sett opp iterasjons-skjemaet og kjør 2 iterasjoner med de to startverdiene $x^0 = 3.0$ og $x^0 = 0.1$. Hva finner du nå? Forklar hvorfor resultatet er forskjellige fra fikspunkt-iterasjonene

Fasit: Iterasjons-skjemaet for Newtons metode er

$$x^{k+1} = g_N(x^k) = x^k - \frac{f(x^k)}{f'(x^k)} = x^k - \frac{\sin x^k}{\cos x^k}.$$

For $x^0 = 3.0$, 2-iterasjoner gir

$$x^1 = 3.142546543074 28, \quad x^2 = 3.14159265330048.$$

For $x^0 = 0.1$, 2-iterasjoner gir

$$x^1 = -3.346720854505436e-04, \quad x^2 = 1.249502806993999e-11.$$

Vi ser at vi finner $x_2 = \pi$ ved startverdien $x^0 = 3.0$, og vi finner $x_1 = 0$ ved startverdien $x^0 = 0.1$. Dette er fordi Newtons metode konvergerer alltid hvis startverdi ligger nær løsningen.

c) I denne oppgaven skal vi se på feil. La e^k være feilen i iterasjons-steg nr. k, d.v.s., $e^k = |s - x^k|$ hvor s er eksakt løsningen.

Man kan vise at, (du skal IKKE vise det!) for fikspunkt iterasjon har vi lineær konvergens, d.v.s.,

$$e^{k+1} \leq m e^k, \quad \text{hvor} \quad m = \max |g_F'(x)|,$$

mens for Newtons metode har vi kvadratisk konvergens, d.v.s.,

$$e^{k+1} \leq M(e^k)^2, \quad \text{hvor} \quad M = \frac{\max |f''(x)|}{2 \min |f'(x)|}.$$

og $g_N(x) = x - f(x)/f'(x)$ er iterasjonsfunksjonen for Newtons metode.

Sett $x^0 = 3.0$ for begge metoder. Hvor mange iterasjoner trengs for fikspunkt iterasjon og Newtons metode, slik at vi er garantert en feil på mindre enn $\varepsilon = 10^{-10}$? Kommenter resultatet.

Fasit: For fikspunk iterasjon, ut fra feilestimatoren $e^{k+1} \leq m e^k$ kan man lett utlede at

$$e^k \leq m^k e^0.$$

Siden $g_F'(x) = 1 + \cos x$, derfor for x ligger mellon 3 og π, er $|g'| < 1 + \cos 3 = 0.011$. Da kan vi bruke $m = 0.011$. Videre er $e^0 = \pi - 3 < 0.15$. Kravet $e^k \leq \varepsilon$ gir

$$k \geq \frac{\ln e^0 - \ln \varepsilon}{-\ln m} = 4.685,$$

d.v.s., vi må kjøre minst 5 iterasjoner.

For Newtons metode, skal vi først finne M

$$M = \frac{|\sin(3)|}{2|\cos(3)|} = 0.071.$$

Vi har også $e^0 < 0.15$.

Ut fra feilestimatoren til Newtons metoden $e^{k+1} \leq M(e^k)^2$, har vi

$$\begin{aligned}
e^1 &\leq M(e^0)^2 \\
e^2 &\leq M(e^1)^2 \leq M(M(e^0)^2)^2 = M^{1+2}(e^0)^{2\times 2} \\
e^3 &\leq M(e^2)^2 \leq M(M^{1+2}(e^0)^{2\times 2})^2 = M^{1+2+2^2}(e^0)^{2\times 2\times 2}
\end{aligned}$$

Ved induksjon kan man lett vise at

$$e^k \leq M^{1+2+\cdots+2^{k-1}}(e^0)^{2^k} = M^{2^k-1}(e^0)^{2^k} = (Me^0)^{2^k}/M.$$

Kravet $e^k \leq \varepsilon$ gir

$$(Me^0)^{2^k} \leq M\varepsilon, \quad \rightarrow \quad 2^k \geq \ln(M\varepsilon)/\ln(Me^0) \quad \rightarrow \quad k \geq \ln\left(\frac{\ln(M\varepsilon)}{\ln(Me^0)}\right)/\ln(2).$$

Sett inn verdiene for M, e^0 og ε, finner vi

$$k \geq 2.499$$

d.v.s., vi trenger minst 3 iterasjoner.

Vi ser at Newtons metode har kvadratisk konvergens, derfor konvergerer metoden raskere enn fikspunkt iterasjon.

Oppgave 4 La $u(x,t)$ være løsningen til adveksjon-diffusjons likningen

$$\begin{aligned} &u_t + au_x = bu_{xx}, \qquad (0 \le x \le 1,\ t \ge 0) \\ &u(0,t) = 0, \quad u(1,t) = 0, \qquad (t \ge 0) \\ &u(x,0) = \bar{u}(x), \qquad (0 < x < 1) \end{aligned}$$

Her er a, b positive konstanter, $a > 0$, $b > 0$.

Vi ønsker å finne numeriske løsninger til differensiallikningen. La u_i^n være den numeriske tilnærmelsen til $u(x_i, t_n)$ hvor $x_i = i \cdot h$, $t_n = n \cdot k$, og h og k er gitte størrelser i x og t retninger til et uniformt gitter. Vi diskretiserer i x retningen med sentral differenser.

a) Bruk forlengs Euler i tidsdiskretiseringen og sett opp et eksplisitt numerisk skjema.

Vis at under følgende stabilitetsbetingelsene

$$k \le h^2/(2b), \qquad \text{og} \qquad h \le 2b/a$$

oppfyller skjemaet maksimumsprinsippet, d.v.s.,

$$\max_i \left|u_i^{n+1}\right| \le \max_i |u_i^n| .$$

Fasit: Eksplisitt skjema ser som ut:

$$\frac{u_i^{n+1} - u_i^n}{k} + a \cdot \frac{u_{i+1}^n - u_{i-1}^n}{2h} = b \cdot \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2}$$

eller etter rydding

$$u_i^{n+1} = \left(\frac{ak}{2h} + \frac{bk}{h^2}\right) u_{i-1}^n + \left(1 - \frac{2bk}{h^2}\right) u_i^n + \left(\frac{bk}{h^2} - \frac{ak}{2h}\right) u_{i+1}^n .$$

Nå skal vi sjekke stabilitetsbetingelsene. Ta absolut verdi på begge side og bruk trekant ulikhet:

$$\begin{aligned} \left|u_i^{n+1}\right| &= \left|\left(\frac{ak}{2h} + \frac{bk}{h^2}\right) u_{i-1}^n + \left(1 - \frac{2bk}{h^2}\right) u_i^n + \left(\frac{bk}{h^2} - \frac{ak}{2h}\right) u_{i+1}^n\right| \\ &\le \left|\left(\frac{ak}{2h} + \frac{bk}{h^2}\right)\right| \left|u_{i-1}^n\right| + \left|\left(1 - \frac{2bk}{h^2}\right)\right| |u_i^n| + \left|\left(\frac{bk}{h^2} - \frac{ak}{2h}\right)\right| \left|u_{i+1}^n\right| \end{aligned}$$

Bytt ut $|u_i^n|$, $\left|u_{i-1}^n\right|$ og $\left|u_{i+1}^n\right|$ med den største de kan være, kan vi beholde ulikheten

$$\left|u_i^{n+1}\right| \le \left|\left(\frac{ak}{2h} + \frac{bk}{h^2}\right)\right| \max_i |u_i^n| + \left|\left(1 - \frac{2bk}{h^2}\right)\right| \max_i |u_i^n| + \left|\left(\frac{bk}{h^2} - \frac{ak}{2h}\right)\right| \max_i |u_i^n| .$$

Under betingelsene $2bk \le h^2, h \le 2b/a$, er alle konstantene i absolutverdi-tegn positive, og vi har

$$\left|u_i^{n+1}\right| \le \left[\left(\frac{ak}{2h} + \frac{bk}{h^2}\right) + \left(1 - \frac{2bk}{h^2}\right) + \left(\frac{bk}{h^2} - \frac{ak}{2h}\right)\right] \max_i |u_i^n| = \max_i |u_i^n| .$$

Siden ulikheten gjelder for alle i, gjelder den også for i når $\left|u_i^{n+1}\right|$ har maksimum verdi. Derfor har vi

$$\max_i \left|u_i^{n+1}\right| \le \max_i |u_i^n|$$

som er maksimumsprinsippet.

b) Bruk baklengs Euler i tidsdiskretiseringen og sett opp et implisitt numerisk skjema.

Vis at skjemaet oppfyller maksimumsprinsippet hvis $h < 2b/a$. Forklar hvorfor denne betingelsen er mye bedre enn det tilsvarende for det eksplisitte skjemaet.

Fasit: Implisitt skjema ser som ut:

$$\frac{u_i^{n+1} - u_i^n}{k} + a \cdot \frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2h} = b \cdot \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{h^2}$$

eller etter rydding

$$-\left(\frac{ak}{2h} + \frac{bk}{h^2}\right) u_{i-1}^{n+1} + \left(1 + \frac{2bk}{h^2}\right) u_i^{n+1} + \left(\frac{ak}{2h} - \frac{bk}{h^2}\right) u_{i+1}^{n+1} = u_i^n.$$

Her trenger man å løse et tridiagonal lineært likningssystem i hvert tidssteg.

Skjemaet kan også skrives som

$$\left(1 + \frac{2bk}{h^2}\right) u_i^{n+1} = u_i^n + \left(\frac{ak}{2h} + \frac{bk}{h^2}\right) u_{i-1}^{n+1} + \left(\frac{bk}{h^2} - \frac{ak}{2h}\right) u_{i+1}^{n+1}.$$

Under betingelsen $h \leq 2b/a$ er $\frac{bk}{h^2} - \frac{ak}{2h} \geq 0$. Vi tar absolutverdi på begge sider og deretter bruker trekant ulikhet:

$$\begin{aligned}
\left(1 + \frac{2bk}{h^2}\right) \left|u_i^{n+1}\right| &\leq \left|u_i^n\right| + \left(\frac{ak}{2h} + \frac{bk}{h^2}\right) \left|u_{i-1}^{n+1}\right| + \left(\frac{bk}{h^2} - \frac{ak}{2h}\right) \left|u_{i+1}^{n+1}\right| \\
&\leq \max_i \left|u_i^n\right| + \left(\frac{ak}{2h} + \frac{bk}{h^2}\right) \max_i \left|u_i^{n+1}\right| + \left(\frac{bk}{h^2} - \frac{ak}{2h}\right) \max_i \left|u_i^{n+1}\right| \\
&= \max_i \left|u_i^n\right| + \frac{2bk}{h^2} \max_i \left|u_i^{n+1}\right|.
\end{aligned}$$

Siden ulikheten gjelder for alle i, gjelder den også når vi tar maksimum på venstre side. Vi har:

$$\left(1 + \frac{2bk}{h^2}\right) \max_i \left|u_i^{n+1}\right| \leq \max_i \left|u_i^n\right| + \frac{2bk}{h^2} \max_i \left|u_i^{n+1}\right|.$$

som gir maksimumsprinsippet

$$\max_i \left|u_i^{n+1}\right| \leq \max_i \left|u_i^n\right|.$$

Vi ser at betingelsen $h \leq 2b/a$ er felles for begge skjema, som egentlig ikke er en streng betingelse. Den bare krever at roms-intervallet må være mindre enn en konstant $2b/a$. Men for eksplisitt skjema, kreves i tillegg at $k \leq h^2/(2b)$, som setter strengt krav på tids-intervallet. Vi ser at hvis h er lite, er h^2 mye mindre, derfor må vi bruke en veldig lite k og mange tidsteg for å få garantert stabiliteten. Dette er et uønskeligt krav. Implisitt skjema unngår dette kravet, og man kan velge stor k. Men i hvert tidssteg må man løse et tridiagonal lineært likningssystem.

c) Skriv et Matlab program som beregner den numeriske løsningen til differensiallikningen, med både eksplisitt og implisitt skjema. La $u(x, 0) = \bar{u}(x) = \sin(4\pi x)$. Programmet skal sjekke stabilitetsbetingelsene og gi melding hvis dette ikke er oppfylt. Den skal også plotte løsningen.

Programmet kan begynne med:

```
function u=adv_diff(a,b,T,h,k,metode)
% function u=adv_diff(a,b,T,h,k,metode)
% loesninger av adveksjon-diffusjons likningen ved numeriske metoder
% input parameter:
%  a, b: koeffisienter til diff.liknignen
%  T:  tiden vi skal beregne loesningen, t=T.
%  h:  gitter stoerrelse i x
%  k:  gitter stoerrelse i t
%  metode: hvilken metode skal brukes.
%          hvis metode==1, bruk eksplisitt metode,
%          hvis metode==2, bruk implisitt metode.
%
%  resultat:
%        u:  numerisk loesning i tidspunkt t=T.
```

Fasit: Et forslag for Matlab programmet er gitt:

```
function u=adv_diff(a,b,T,h,k,metode)
% function u=adv_diff(a,b,T,h,k,metode)
% loesninger av adveksjon diffusjons likningen ved numeriske metoder
% input parameter:
%  a, b: koeffisienter til diff.liknignen
%  T:  tiden vi skal beregne loesningen, t=T.
%  h:  gitter stoerrelse i x
%  k:  gitter stoerrelse i t
%  metode: hvilken metode skal brukes.
%          hvis metode==1, bruk eksplisitt metode,
%          hvis metode==2, bruk implisitt metode.
%
%  resultat:
%        u:  numerisk loesning i tidspunkt t=T.

N = 1/h; % antall interval i x
M = T/k; % antall interval i t
x=[0:h:1]';
up=zeros(size(x));
u=zeros(size(x));
up=sin(4*pi*x); % sett initial data

m1=a*k/2/h;  m2=b*k/h/h;

if (metode==1) % bruk eksplisitt skjema
  if ((m1>m2) | (2*m2>1)) % sjekk stabilitetsbetingelsene
    disp('Warning: stabil.bet. ikke oppfylt for eksplisitt skjema!')
  end
  for n=1:1:M,
    for i=2:1:N,
  u(i) = (m1+m2)*up(i-1) + (1-2*m2)*up(i) + (m2-m1)*up(i+1);
    end
    up=u;
  end
```

```
  plot(x,u), title('Loesning med eksplisitt metode'),

elseif (metode==2) % bruk implisitt skjema
  if (m1>m2) % sjekk stabilitetsbetingelsene
    disp('Warning: stabilitetsbet. ikke oppfylt for implisitt skjema!')
  end
  % sett opp ligningssystemet
  A = zeros(N-1,N-1);
  d=1+2*m2; % diagonal
  d1=m1-m2; % upper diag
  d2=-m1-m2; % lower diag
  A=diag(ones(N-1,1)*d) + diag(ones(N-2,1)*d1,1) + ...
  diag(ones(N-2,1)*d2,-1); % tridiagonal matrise
  for n=1:1:M,
    u(2:N)=A\up(2:N);
    up=u;
  end
  plot(x,u), title('Loesning med implisitt metode'),
end
```