UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i	INF-MAT2350 — Numeriske Beregninger
Eksamensdag:	14 juni 2004
Tid for eksamen:	9.00 – 12.00
Oppgavesettet er på 5 sider.	
Vedlegg:	Ingen
Tillatte hjelpemidler:	ingen

Kontroller at oppgavesettet er komplett før
du begynner å besvare spørsmålene.

Alle 6 delspørsmål vektlegges likt.

Oppgave 1 Monte Carlo

Vi skal utvikle en Monte Carlo algoritme for å estimere volumet til ellipsoiden

$$E = \left\{ (x,y,z) : x^2 + \frac{y^2}{4} + \frac{z^2}{9} = 1 \right\}.$$

Det kan vises at $E \subset B$ hvor B er boksen

$$B = \{(x,y,z) : -1 \le x \le 1, -2 \le y \le 2, -3 \le z \le 3\}.$$

Forklar hvordan vi kan bruke matlabfunksjonen `rand` (se appendiks) til å estimere volumet av E. Lag et matlabskript som kaller `rand` et visst antall ganger $3m$ og skriver ut et estimat for volumet av E.

Svar: Siden rand gir et tlfeldig tall i intervallet [0,1] vil -h+2*h*rand gi et tilfeldig tall i intervallet [-h,h]. Vi kan generere m punkter i B ved å kalle rand 3m ganger som i programmet nedenfor. Et estimat på volumet får vi ved å finne hvor mange av de m punktene som ligger i ellipsoiden og gange dette med volumet av B.

```
m = 1000000;
NumberInside  = 0;
for k=1:m
  x = -1+2*rand;
  y = -2+4*rand;
  z = -3+6*rand;
  if x*x + y*y/4.0 + z*z/9.0 <= 1
    NumberInside = NumberInside + 1;
```

(Fortsettes på side 2.)

```
  end
end
sprintf('Volume = %8.3f', 48 * NumberInside / m);
```

Oppgave 2 Splines

La

$$a = x_1 < \cdots < x_{n+1} = b$$

være en partisjon av intervallet $[a, b]$. En spline $s : [a, b] \to \mathbb{R}$ av orden $k \geq 1$ (grad $k-1$) over denne partisjonen kan på hvert delintervall $[x_i, x_{i+1}]$ skrives på formen

$$s(z) = \rho_{i,k} + \rho_{i,k-1}(z - x_i) + \cdots + \rho_{i,1}(z - x_i)^{k-1},$$

for $i = 1, 2, \ldots, n$.

2a

La ρ være en koeffisientmatrise av dimension $n \times k$ og la $x = (x_1, \ldots, x_{n+1})$ være en vektor som ovenfor. Skriv en matlabfunksjon `splineplot(x,rho)` som plotter splinen s på $[a, b]$. Du skal bruke matlabrutinene `mkpp` og `ppval` som er gitt i appendikset.

Svar:

```
function splineplot(x,rho)
s = mkpp(x,rho);
z = linspace(x(1),x(n+1));
svals = ppval(s,z);
plot(z,svals);
```

2b

Anta at $f : [a, b] \to \mathbb{R}$ er en deriverbar funskjon. Vi ønsker å plotte en approksimasjon til den første deriverte f' gitt bare verdiene av f i noen punkter. Anta at punktene er

$$a = x_1 < \cdots < x_{n+1} = b.$$

Matlabrutinen `spline(x,y)` finner en kubisk spline s slik at $s(x_i) = y_i$, $i = 1, \ldots, n+1$. Skriv en matlab funksjon som plotter approksimasjonen s' til f' hvor $s : [a, b] \to \mathbb{R}$ er splinen slik at $s(x_i) = f(x_i)$, $i = 1, \ldots, n+1$. Du skal bruke den innebygde matlabrutinen `unmkpp` i appendikset.

Svar:

```
y =eval(fname,x);
s = spline(x,y);
[x1,rho,n1,k1] = unmkpp(s);
drho = [3*rho(:,1) 2*rho(:,2) \rho(:,3)];
splineplot(x,drho);
```

(Fortsettes på side 3.)

Oppgave 3 Numerisk integrasjon

La f være en funksjon med to kontinuerlige deriverte på intervallet $[-1, 1]$. Vi skal approksimere integralen til f med midpunkt-regelen

$$\int_{-1}^{1} f(x)\,dx \approx 2f(0).$$

3a

Vis at regelen er eksakt for polynomer av grad ≤ 1.

Svar:

Siden både integralen og midpunkt-regelen er lineære operatorer er det nok å vise at regelen er eksakt for polynomene 1 og x. Siden

$$\int_{-1}^{1} 1\,dx = 2, \qquad \int_{-1}^{1} x\,dx = 0,$$

ser vi at regelen er eksakt for disse to polynomene.

3b

Ved å bruke en Taylor rekke, vis at

$$\left| \int_{-1}^{1} f(x)\,dx - 2f(0) \right| \leq \|f''\|/3,$$

hvor

$$\|f''\| := \max_{-1 \leq x \leq 1} |f''(x)|.$$

Svar:

Vi integrerer Taylor-rekken

$$f(x) = f(0) + xf'(0) + \frac{x^2}{2} f''(\xi_x),$$

hvor ξ_x er mellom 0 og x, og dermed i $[-1, 1]$, og får

$$\int_{-1}^{1} f(x)\,dx = 2f(0) + \int_{-1}^{1} \frac{x^2}{2} f''(\xi_x)\,dx.$$

Dermed er

$$\left| \int_{-1}^{1} f(x)\,dx - 2f(0) \right| \leq \|f''\| \int_{-1}^{1} \frac{x^2}{2}\,dx = \|f''\|/3.$$

(Fortsettes på side 4.)

Oppgave 4 Newton's metode

Vi ønsker å finne et nullpunkt til to deriverbare funskjoner $f : \mathbb{R}^2 \to \mathbb{R}$ og $g : \mathbb{R}^2 \to \mathbb{R}$, d.v.s. et punkt (x, y) hvor

$$f(x, y) = g(x, y) = 0.$$

Utled Newton's metode for dette problemet.

Svar: Gitt en tilnærmelse (x_k, y_k) til (x, y), bergegner vi en ny tilnærmelse (x_{k+1}, y_{k+1}) ved følgende lineærisering:

$$f(x_{k+1}, y_{k+1}) \approx f(x_k, y_k) + s_k f_x(x_k, y_k) + t_k f_y(x_k, y_k) = 0,$$

$$g(x_{k+1}, y_{k+1}) \approx g(x_k, y_k) + s_k g_x(x_k, y_k) + t_k g_y(x_k, y_k) = 0,$$

hvor

$$s_k = x_{k+1} - x_k, \qquad t_k = y_{k+1} - y_k.$$

Løser vi dette med hensyn til s_k og t_k får vi

$$Jf(x_k, y_k) \begin{pmatrix} s_k \\ t_k \end{pmatrix} = - \begin{pmatrix} f(x_k, y_k) \\ g(x_k, y_k) \end{pmatrix},$$

hvor $Jf(x_k, y_k)$ er Jakobimatrisen

$$Jf(x_k, y_k) = \begin{pmatrix} f_x(x_k, y_k) & f_y(x_k, y_k) \\ g_x(x_k, y_k) & g_y(x_k, y_k) \end{pmatrix}.$$

Dermed får vi

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} - Jf(x_k, y_k)^{-1} \begin{pmatrix} f(x_k, y_k) \\ g(x_k, y_k) \end{pmatrix},$$

hvis matrisen $Jf(x_k, y_k)$ er inverterbar.

Appendiks

```
 RAND   Uniformly distributed random numbers.
    RAND(N) is an N-by-N matrix with random entries, chosen
    from a uniform distribution on the interval (0.0,1.0).
    RAND(M,N) and RAND([M,N]) are M-by-N matrices with
    random entries. RAND(M,N,P,...) or RAND([M,N,P,...])
    generate random arrays. RAND with no arguments is a
    scalar whose value changes each time it is referenced.
    RAND(SIZE(A)) is the same size as A.

MKPP Make piecewise polynomial.
    PP = MKPP(BREAKS,COEFS) puts together a piecewise
    polynomial PP from its breaks and coefficients.
    BREAKS must be a vector of length L+1 with strictly
    increasing elements representing the start and end
    of each of L intervals.  The matrix COEFS must be
    L-by-K with each row COEFS(i,:) representing the
    coefficients of the order K polynomial on the interval
    [BREAKS(i),BREAKS(i+1)].
```

(Fortsettes på side 5.)

```
 PPVAL  Evaluate piecewise polynomial.
    V = PPVAL(PP,XX) returns the value at the points XX
    of the piecewise polynomial contained in PP, as
    constructed by SPLINE or the spline utility MKPP.

UNMKPP Supply details about piecewise polynomial.
    [BREAKS,COEFS,L,K] = UNMKPP(PP) extracts from
     the piecewise polynomial PP its breaks, coefficients,
     number of pieces, and order of its target. PP would
     have been created by SPLINE or the spline utility MKPP.
```

Lykke til!