UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i	MoD 190 — Numeriske beregninger
Eksamensdag:	31. Mai 2002
Tid for eksamen:	9.00 – 13.00
Oppgavesettet er på 4 sider.	
Vedlegg:	1
Tillatte hjelpemidler:	Ingen

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

Alle 8 delspørsmål vektlegges likt.

Oppgave 1 Diverse Matlab-oppgaver

1a

Skriv opp vektoren z etter at følgende Matlab-kommandoer er utført:

```
A = [3 2 1 ; 2 3 2 ; 1 2 3]
x = A(:,1).*A(:,3)
y = A(2,:)
B = x*y
z = [B(1,:) B(2,:) B(3,:)]
z(1:2:9) = ones(1,5)
```

Vis resultatet etter hver beregning.

1b

Skriv en Matlab funksjon `Pellipse(P,A,theta)` som plotter en skjev ellipse gitt ved

$$\begin{aligned}
x(t) &= \cos(\theta)\left[\frac{P-A}{2}+\frac{P+A}{2}\cos(t)\right]-\sin(\theta)\left[\sqrt{A\cdot P}\sin(t)\right] \\
y(t) &= \sin(\theta)\left[\frac{P-A}{2}+\frac{P+A}{2}\cos(t)\right]+\cos(\theta)\left[\sqrt{A\cdot P}\sin(t)\right]
\end{aligned}$$

for $0 \le t \le 2\pi$. Din implementasjon skal ikke ha noen løkker.

(Fortsettes på side 2.)

1c

Skriv et Matlab program som lager en tabell over verdien av integralene

$$\int_0^{k/10} \sqrt{x} e^{-x} dx, \quad \text{for } k = 1, 2, \ldots, 50.$$

Bruk Matlabrutinen `quad` med toleranse 10^{-6} (se vedlegg). Hvorfor kan man forvente at `quad` vil trenge mange evalueringer av integranden når x er nær 0? Prøv å organisere beregningene slik at programmet ikke foretar unødvendig mange beregninger av integranden.

1d

Et polynom på formen

$$p(x) = a_1 + a_2 x^2 + a_3 x^4 + \cdots + a_n x^{2n-2}$$

sies å være *like*, mens et polynom på formen

$$p(x) = a_1 x + a_2 x^3 + a_3 x^5 + \cdots + a_n x^{2n-1}$$

sies å være *odde*. Generaliser `HornerV(a,z)` (se vedlegg) slik at funksjonen har et valgfritt tredje argument `type` som indikerer om det underliggende polynomet er like eller odde. Ved et kall på `HornerV(a,z,'like')` antas det at a_k er koeffisienten til x^{2k-2}, mens det ved et kall på `HornerV(a,z,'odde')` antas at a_k er koeffisienten til x^{2k-1}. Test på antall argumenter ved å bruke verdien av den innebygde Matlabvariablen `nargin`.

Oppgave 2 En ikke-lineær ligning

2a

La $t > 1$ være gitt. Forklar hvorfor ligningen

$$f(x) = e^x + t(x-1) = 0 \tag{1}$$

har en entydig løsning $x = x(t)$ med $0 < x(t) < 1$.

2b

Vi benytter Newton's metode med $x_0 = 1$ til å løse ligningen (1). (Du skal ikke lage et Matlab program for Newtons metode). La $\{x_k\}$ være følgen generert ved Newton's metode og sett $e_k = x_k - x$ for alle k, hvor $x = x(t)$. Vis for $k = 0, 1, 2, \ldots$ at

$$e_{k+1} = e_k - \frac{f(x_k)}{f'(x_k)} \tag{2}$$

$$0 = f(x_k) - e_k f'(x_k) + \frac{1}{2} e_k^2 f''(\xi_k) \tag{3}$$

$$e_{k+1} = \frac{1}{2} e_k^2 \frac{f''(\xi_k)}{f'(x_k)} \tag{4}$$

(Fortsettes på side 3.)

for en ξ_k mellom x og x_k.

2c

Vis at følgen $\{x_k\}$ generert ved Newton's metode tilfredsstiller

$$0 < x(t) < x_{k+1} < x_k \leq 1, \quad \text{for} \quad k = 0, 1, 2, \ldots$$

2d

Anta vi blir bedt om å lage en tabell over numeriske tilnærmelser til $x(t_n)$ for $t_n = 1 + nh$ for $n = 0, 1, \ldots, 1000$. Det blir foreslått at tilnærminger y_n til $x(t_n)$ skal beregnes ved hjelp av algoritmen

$y_0 = 0$

for $n = 0 : 999$

$y_{n+1} = y_n + h\frac{1-y_n}{t_n+e^{y_n}}$

Hva er begrunnelsen for denne algoritmen? Hint: $x(t)$ er løsning av et initialverdiproblem.

Lykke til!

(Fortsettes på side 4.)

Matlab-vedlegg

```
  function pVal = HornerV(a,z)
% pVal = HornerV(a,z)
% evaluates the Vandermonde interpolant on z where
% a is an n-vector and z is an m-vector.
%
% pVal is a vector the same size as z with the property that if
%
%          p(x) = a(1) + .. +a(n)x^(n-1)
% then
%          pVal(i) = p(z(i))  ,  i=1:m.

n = length(a);
m = length(z);
pVal = a(n)*ones(size(z));
for k=n-1:-1:1
   pVal = z.*pVal + a(k);
end
```

```
QUAD   Numerically evaluate integral, adaptive Simpson quadrature.
    Q = QUAD(FUN,A,B) tries to approximate the integral of function
    FUN from A to B to within an error of 1.e-6 using recursive
    adaptive Simpson quadrature.  The function Y = FUN(X) should
    accept a vector argument X and return a vector result Y, the
    integrand evaluated at each element of X.
```