Matematik
Chalmers tekniska högskola 2019-06-03 kl. 08:30-12:30

Tentamen MVE355, Programmering och numeriska beräkningar med Matlab.

Ansvarig: Katarina Blom , tel 772 10 97.
Telefonvakt under tentan: Katarina Blom, ankn 1097 Plats: L
Inga hjälpmedel. Kalkylator ej tillåten.

Betygsgränser: 16-23 p. ger betyget 3, 24-31 p. ger betyget 4 och 32 p. eller mer ger betyget 5. Maxpoäng är 40.
Lösningar kommer att läggas ut på kurshemsidan första arbetsdagen efter tentamenstillfället. Resultat meddelas via epost från LADOK.

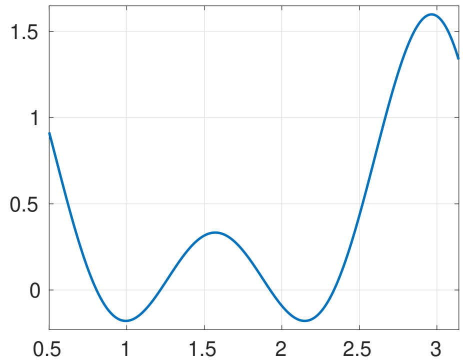
---

1 Matlabsekvensen till vänster nedan ritar grafen till till funktionen

$$f(x) = \sin(3x) + \cos(4x) + \frac{1}{3}$$

på intervallet $\frac{1}{2} \leq x \leq \pi$. Figuren visas till höger nedan.

```
x = 1/2:0.01:pi;
y = sin(3*x) + cos(4*x)+1/3;
plot(x,y);
grid on
```



(a) Skriv ett program som bestämmer det största y-värdet som ritas i figuren (dvs. det största y-värdet som används i programsekvensen som ritar grafen ovan). Låt programmet även bestämma x-värdet som hör till det största y-värdet. Använd t.ex. kommandot `max`. (3p)

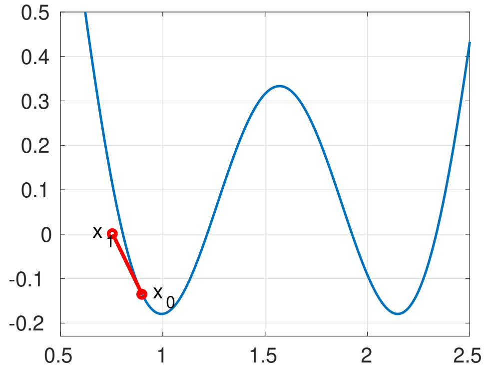
(b) Funktionen har ett antal nollställen på intervallet $\frac{1}{2} \leq x \leq \pi$. Skriv en sekvens i Matlab som bestämmer nollstället längst till höger och markerar det med en liten ring i figuren. Använd `fzero`. (4p)

(c) Om man använder Newton's metod med startvärdet $x_0 = 0.9$ kommer man att få konvergens mot ett av nollställena i figuren. Vilket av nollställena? (Ledning: $f'(0.9) \approx -0.9$). Motivera svaret ordentligt. (Rita gärna en figur). (2p)

(a)
```
[ymax, p] = max(y);
xmax = x(p);
svar = [ymax, xmax]
```

(b)
```
f=@(x)sin(3*x) + cos(4*x)+1/3;
x0=fzero(f,2.5);
plot(x0,f(x0),'o')
```

(c) Man kommer att bestämma det minsta nollstället. Newton's metod följer hela tiden tangenten till den skär x-axeln. I figuren nedan har man ritat tangenten mellan $(x_0, f(x_0)) = (0.9, f(0.9))$ och $(x_1, 0)$ där $x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$



---

2 Låt $f(x) = \sin(3x) + \cos(4x) + \dfrac{1}{3}$ (samma funktion som i uppgift 1)

(a) Skriv en sekvens i Matlab som beräknar en approximation av integralen (4p)

$$\int_{1/2}^{\pi} f(x)dx$$

Använd mittpunktsregeln och dela in integrationsintervallet i 30 delar.

(b) Låt `qm` vara integralberäkningen med mittpunktsregeln från (a). Som bekant gäller (3p)

$$\int_{1/2}^{\pi} f(x)dx = \texttt{qm} + \frac{\pi - 1/2}{24} f''(\xi)h^2$$

där $1/2 \leq \xi \leq \pi$ och h steglängden (som användes i (a)). Skriv en sekvens i Matlab som bestämmer ett värde på ξ. Använd `integral` för att beräkna $\int_{1/2}^{\pi} f(x)dx$. (Ledning: Ett värde på ξ finns på intervallet $[1, 1.5]$).

(a)
```
n = 30; a = 0.5; b = pi; h = (b-a)/n;
x = linspace(a,b,n+1);
f=@(x)sin(3*x) + cos(4*x)+1/3;
qm = sum(h*f(x(1:n)+h/2));
disp(qm);
```

(b)
```
f=@(x)sin(3*x) + cos(4*x)+1/3;
ddf=@(x)-16*cos(4*x)-9*sin(3*x); % 2:a derivatan
e = integral(f,1/2,pi);
g=@(x)24*(e-qm)/((pi-1/2)*h^2)-ddf(x);
xsi=fzero(g,[1,1.5]);
```

---

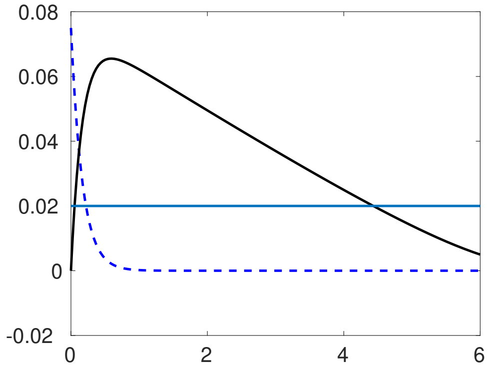
3 (a) Formulera Eulers metod för lösning av differentialekvationer. (3p)

(b) Differentialekvationen nedan beskriver hur mängden av en viss drog i kroppen varierar över tid. I ekvationen är $x(t)$ mängden av drogen i mag-tarmkanalen, $y(t)$ är mängden av drogen i blodet och t är tiden i timmar. (3p)

$$\begin{cases} x'(t) = -6x(t) \\ y'(t) = 6x(t) - \dfrac{0.014y(t)}{y(t) + 0.005} \end{cases}$$

Skriv ett program i Matlab som löser differentialekvationen på intervallet $0 \leq t \leq 6$. Låt $x(0) = 0.07$ och $y(0) = 0$. Använd `ode45`.

(c) I figuren har man använt samma begynnelsevärden som i (b) och ritat lösningskurvorna för $x(t)$ och $y(t)$ på intervallet $0 \leq t \leq 6$. Man har även ritat en linje som markerar 0.02. Skriv en sekvens i Matlab som skriver ut hur lång tid det tar för drogen i blodet att understiga 0.02. (3p)



(d) Skriv ytterligare ett program i Matlab som löser begynnelsevärdesproblemet i (b). Använd Eulers metod och steglängd $h = 0.1$. (3p)

(a) Se litteraturen

(b)
```
f=@(t,u)[-6*u(1);6*u(1)-0.014*u(2)/(u(2)+0.05)];
[t,U]=ode45(f,[0,6],[0.075;0]);
```

(c)
```
p=max(find(U(:,2)>0.02));
svar = t(p)
```

(d)
```
f=@(t,u)[-6*u(1);6*u(1)-0.014*u(2)/(u(2)+0.05)];
a=0; b=6; h = 0.1; N = (b-a)/h;
t=a+(0:N)*h; U=zeros(2,length(t));
U(:,1)=[0.075;0];
for n = 1:N
  U(:,n+1) = U(:,n)+h*f(t(n),U(:,n));
end
```

---

4 En $n \times n$ matris med bara 0:or över huvuddiagonalen kallas vänstertriangulär. Matriserna $\boldsymbol{A}$ och $\boldsymbol{B}$ nedan är t.ex. vänstertriangulära.

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 5 & 1 \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 9 & 1 & 2 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

(a) Skriv en kod i Matlab som verifierar att matriserna $\boldsymbol{A}$ och $\boldsymbol{B}$ ovan är vänstertriangulära. (Du får använda dig av svaret i (b)-uppgiften om du vill, även om du inte löst uppgiften). (4p)

(b) Skriv en funktion (5p)

```
function t = vtriang(A)
```

som returnerar `true` om en matris `A` är vänstertriangulär (och `false` om matrisen inte är vänstertriangulär). (Du kan anta att matrisen som funktionen anropas med är kvadratisk, dvs. har lika många rader och kolumner)

(c) En matris kallas permuterat vänstertriangulär[1] om den kan göras vänstertrian- (3p) gulär med radbyten. T.ex. matriserna $\boldsymbol{C}$ och $\boldsymbol{D}$ är permuterat vänstertriangulära

$$\boldsymbol{C} = \begin{bmatrix} 0 & 5 & 1 \\ 3 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \boldsymbol{D} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 9 & 1 & 2 & 0 \end{bmatrix}$$

Om man byter plats på första och sista raden i $\boldsymbol{C}$ får man matrisen $\boldsymbol{A}$ ovan. Genom att byta plats på de två sista raderna i $\boldsymbol{D}$ blir den vänstertriangulär. Skriv en funktion som returnerar `true` om en matris är permuterat vänstertriangulär. Låt funktionen ha matrisen som inparameter. Du kan utgå från att funktionen bara anropas med kvadratiska matriser.

(a)
```
A = [1 0 0; 3 1 0; 0 5 1];
B = [1 0 0 0; 0 0 0 0; 9 1 2 0; 0 1 0 1];
disp(vtriang(A));
disp(vtriang(B));
```

(b)
```
function t = vtriang(A)
  [m,n] = size(A);
  for i = 1:n
    for j = i+1:n
      if A(i,j)~=0
        t = false;
        return
      end
    end
  end
```

(c)
```
function t = pvtriang(A)
  for i = n:-1:1
    nsk(i) = nollisekvens(A(i,:));
  end
  nsk = sort(nsk);
  B = A(nsk,:);
  t = vtriang(B);
```

där funktionen `nollisekvens` räknar antalet nollor i sekvens (från höger till vänster) i en vektor.

```
function nsk = nollisekvens(v)
  nsk = 0;
  for i = n:-1:1
    if v(i) == 0
      nsk = nsk+1;
    else
      return;
    end
  end
```

---

[1]Permuted left triangular i Matlabs dokumentation.