

## Tentamen för kursen TME135 Programmering i Matlab för M1

Tid: 11 januari 2012 kl 14:00-18:00

Lärare: Håkan Johansson, mobil: 0739-678 219, kontor: 772 8575

Tillåtna hjälpmedel: P. Jönsson: MATLAB-beräkningar inom teknik och naturvetenskap, Studentlitteratur.

Lärare besöker salen: c:a 15:30 och 17:00.

Formalia: I programhuvudet på respektive m-fil ska du skriva in din tentamenskod (som du fått vid tentamensanmälan). Om du saknar din kod skall du kontakta tentamensvakten.

Programmen du skriver ska vara tydliga, enkla att följa med förklarande kommentarsatser och det skall vara enkelt att göra rimliga mindre ändringar. Du skall arbeta mot mappen C:\\_ \_ EXAM \_ \_ . Under tentamenstiden används endast denna mapp med undermappar. Det är tillåtet att skapa nya filer utöver de som redan finns, de bör i så fall förses med lämpliga programhuvuden. Programfilerna till respektive uppgift finns och skall efter bearbetning sparas i tillhörande mapp uppgift1-uppgift5. Du får bara lämna en lösning till varje uppgift. En delvis löst uppgift kan ge poäng.

När du är klar med tentamen, stäng av Matlab och logga ut från datorn. Omslaget lämnas till vakten.

Rättning: Resultatet anslås senast tisdagen den 31 januari på avdelningen för Dynamik, Inst för Tillämpad mekanik, Hörsalsvägen 7, plan 3. Det kan också ses i studentportalen några dagar senare. Granskning sker torsdag den 2 februari kl 12:30 til 13:00 och tisdag den 7:e februari kl 12:30 til 13:00.

Betygsgränser: Poängantalet för korrekt besvarad/löst uppgift anges inom parentes (p). Betygsgränser för tentamen är:

Betyg U < 16p ;  $16p \leq$  Betyg 3 < 24p ;  $24p \leq$  Betyg 4 < 32p ; Betyg 5  $\geq$  32p.

**LYCKA TILL!**

1. Matriser (6p)

Givet matrisen

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 \\ 10 & 11 & 12 & 13 \end{bmatrix}$$

- a) Skapa matrisen  $\mathbf{C}$  genom att elementvis dividera  $\mathbf{A}$  med  $\mathbf{B}$  ( $C_{ij} = A_{ij}/B_{ij}$ ). (1p)
- b) Skapa radvektorn  $\mathbf{d} = [1 \ 5 \ 9 \ 2 \ 6 \ 10 \ 3 \ 7 \ 11 \ 4 \ 8 \ 12]$  utifrån matrisen  $\mathbf{A}$ . (1p)
- c) Skapa en vektor  $\mathbf{e}$  utifrån vektorn  $\mathbf{d}$  som innehåller samma element som  $\mathbf{d}$  fast i en slumpmässigt genererad ordning, tex

$$\mathbf{d} = [5 \ 3 \ 9 \ 1 \ 4 \ 11 \ 8 \ 10 \ 2 \ 7 \ 12 \ 6]$$

Använd Matlabs slumpfunktion `randperm`. (2p)

- d) Hitta positionen för värdet 6 i vektorn  $\mathbf{c}$  från föregående uppgift och skriv ut positionen på formen

**Siffran 6 finns på position X i vektorn**

där X är ett heltal (om du ej har lyckats lösa c), använd svaret från b). (2p)

2. Break-even/Skärningspunkt (8p)

Ett företag har tagit fram en produkt som skall ut på marknaden. Utvecklingskostnaden för produkten är 5000kr och produktionskostnaden per enhet antas avta med antalet producerade enheter av produkten ungefär som arcustangens av antalet sålda enheter gånger en konstant. Produkten kommer att säljas för 250kr/st. Detta ger två funktioner,  $f(u)$  som är kostnaden för utveckling och produktion samt  $g(u)$  som är intäkterna för  $u$  st sålda enheter enligt

$$f(u) = 1000 \arctan(0.1u) + 100u + 5000$$

$$g(u) = 250u$$

- a) Eftersom managentpersoner gillar grafer så är din uppgift nu att plotta funktionerna  $f(u)$  och  $g(u)$  i intervallet  $[0 \ 100]$ . Förse grafens axlar med korrekta beskrivningar samt enheter och en passande förklaringsruta (legend). Vidare skall den punkt som ger break-even markeras med en blå asterisk. (4p)
- b) Företagsledningen är nu intresserad av att veta hur många produkter som minst måste säljas för att projektet skall gå med vinst (break-even), det vill säga när intäkterna har passerat kostnaderna. Lös problemet med hjälp av funktionen `fzero`. Skriv ut svaret i kommandofönstret som

Break-even är passerat då XX st enheter har sålts.

där XX är ett heltal. (4p)

- c) Lägg texten från a) som text i figuren

### 3. Taylorutveckling (6p)

Taylorutveckling är en metod som används för att approximera matematiska funktioner kring en punkt med ett polynom. Ett Taylorpolynom av  $n$ :te graden av funktionen  $f(x)$  kring punkten  $a$  skrivs enligt

$$f(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \dots + \frac{f^{(n)}(a)}{n!}(x - a)^n$$

där  $!$  innebär faktoriell (**factorial** i Matlab),  $n$  är antalet termer i utvecklingen och  $f^{(n)}$  är den  $n$ :te derivatan av  $f$ . Denna uppgift går ut på att plotta Taylorutvecklingar av  $\sin(x)$  kring punkten 0 och jämföra med originalfunktionen. Taylorutvecklingen av sinus kring punkten 0 av graden  $m$  ges av

$$\sin x \approx \sum_{n=0}^m \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \dots \quad (1)$$

- Plotta funktionen  $\sin x$  i intervallet  $[-\pi \pi]$ . Kurvan skall vara röd och ha tjockleken 3 punkter. (1p)
- För att göra det lättare att avgöra hur bra approximationen är vill vi bara visa plotten mellan värdena -1.5 och 1.5 i  $y$ -led och mellan  $-\pi$  och  $\pi$  i  $x$ -led. (1p)
- Gör ett program som plottar Taylorutvecklingen av  $\sin x$  kring 0 då  $x \in [-\pi \pi]$  för de första 6 graderna (dvs för  $m = 0, 1, 2, 3, 4, 5$  i ekvation 1). Dessa skall plottas i samma figur som föregående uppgift. Det skall vara enkelt att ändra i programmet så att ett annat antal grader plottas. Varje plot skall ha sin unika färg. (4p)

#### 4. Kurvanpassning (7p)

Vid mätningar av vibrationer kan man mäta hur en viss punkt på en komponent förflyttar sig som funktion av tiden. Vid denna typ av mätningar kommer det i princip alltid med störningar på signalen vilket kan bero på att andra komponenter eller utrustning i närheten stör. För att lättare kunna urskilja en kurva kan man göra ett antagande om hur den ser ut rent matematiskt och därefter försöka anpassa denna matematiska kurvan till mätdata. Till exempel kan man anta att en vibration av något slag har en sinus form, man har då antagit att funktionen som beskriver vibrationen är på formen  $f(t) = C_1 \sin(C_2 t)$  där  $C_1$  och  $C_2$  är okända konstanta koefficienter. Vid kurvanpassning är målet att finna konstanter så att funktionen hamnar så nära den uppmätta signalen som möjligt.

I denna uppgift skall du använda mätdata som finns i filen `Matdata.txt` och därefter anpassa en kurva av formen

$$f(t) = A \sin(t) + Bt^2 \quad (2)$$

För att sedan finna  $A$  och  $B$  skall ett länjart ekvationssystem lösas. Ekvationssystemet sätts upp som

$$\underbrace{\begin{bmatrix} \sin(t_0) & t_0^2 \\ \sin(t_1) & t_1^2 \\ \sin(t_2) & t_2^2 \\ \vdots & \vdots \\ \sin(t_n) & t_n^2 \end{bmatrix}}_{=\mathbf{M}} \underbrace{\begin{bmatrix} A \\ B \end{bmatrix}}_{=\mathbf{x}} = \underbrace{\begin{bmatrix} y(t_0) \\ y(t_1) \\ y(t_2) \\ \vdots \\ y(t_n) \end{bmatrix}}_{=\mathbf{y}} \quad (3)$$

där  $t_0 \dots t_n$  är de tider då mätning har gjorts och  $y(t)$  är signalen vid respektive mätning.

Filen `Matdata.txt` består av en  $1000 \times 4$  matris där den första kolonnen är tiden  $t$ , den andra respektive tredje kolonnen är termerna  $\sin(t)$  och  $t^2$  i ekvation 2 utvärderad i respektive  $t$  (dvs matrisen  $\mathbf{M}$ ) och den fjärde kolonnen är signalen (vektorn  $\mathbf{y}$ ).

- Läs in data från filen `Matdata.txt`. Skapa sedan variablerna `t`, `M` och `y`. `t` är första kolonnen i filen, `M` är andra och tredje och `y` är den fjärde kolonnen. (3p)
- Finn nu de sökta koefficienterna  $A$  och  $B$  genom att lösa ekvationssystemet  $\mathbf{M}\mathbf{x} = \mathbf{y}$ . Ekvationssystemet är överbestämt, men Matlab löser överbestämda system på samma sätt som vanliga system. (1p)
- Skriv ut resultatet på formen

Den anpassade kurvan blir

$$f(t) = XX * \sin(t) + YY * t^2$$

där `XX` och `YY` är flyttal på exponentform (tex `1.0760e-04`) med 4 decimaler (2p)

- Vi vill nu plotta dels mätdata `y` med blå heldragen linje och funktionen i ekvation 2 med de funna koefficienterna med röd linje. (1p)

## 5. Etthundra (13p)

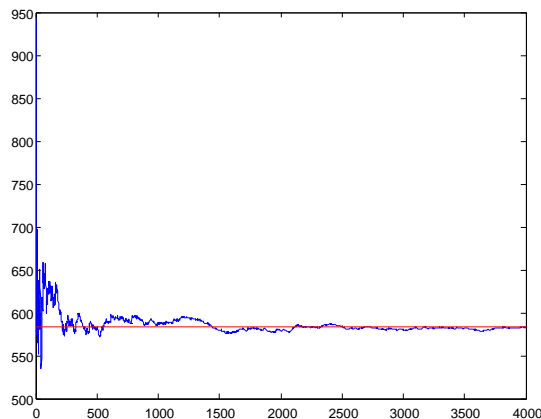
Tärningsspelet etthundra är mycket enkelt. I spelet används endast en sexsidig tärning och när tärningen slås av en spelare får spelaren samma antal poäng som tärningen visar såvida tärningen visar två till sex. Om tärningen visar en etta nollställs poängen. Den spelare som når 100 poäng först vinner. Vi vill undersöka spelet från ett statistiskt perspektiv.

- Skapa en funktion `Etthundra` som kör en omgång Etthundra och returnerar det totala antalet kast som behövdes för att få 100 poäng i rad. Skapa m-filen själv. (5p)
- I `Uppgift5.m`, skriv ett program som kör funktionen `Etthundra` ett förbestämt antal gånger (tex 10) och för varje gång sparar resultat (antalet kast) i en vektor. Det skall vara lätt att ändra antalet funktionsanrop. (3p)
- För att få ett tillförlitligt medelvärde av hur många slag som krävs för att få 100 poäng skall en sk konvergensstudie göras. I detta fall skall det kumulativa medelvärdet  $S$  räknas ut för vektorn  $k$  innehållande resultat från b). Detta ges i form av en vektor med samma längd som antalet kast. Varje element i vektorn innehåller det medelvärde som fås om man räknar med alla kast som förekommer från element 1 till och med detta element. Alltså, det kumulativa medelvärdet  $S(n)$  för element  $n$  i vektorn  $k$  fås av

$$S(n) = \frac{1}{n} \sum_{i=1}^n k(i)$$

Räkna ut det kumulativa medelvärdet för samtliga kast och spara dessa i en vektor. Det kumulativa medelvärdet skall beräknas med elementvisa operationer (en elementvis operation per  $n$ ). Man säger att konvergens är uppnått när förändringen i medelvärdet är försumbart då antalet kast ökar. (3p)

- Plotta det kumulativa medelvärdet med en blå, heldragen linje som funktion av element index. Plotta sedan en horisontell linje över den befintliga blå linjen i samma plot som visar vad det sista medelvärdet är. Denna sista linje skall vara röd och heldragen. (2p)



Figur 1: Exempel på hur resultatet från uppgift d) kan se ut