

## Tentamen för kursen TME135 Programmering i Matlab för M1

Tid: 23 augusti 2010 kl 14:00-18:00

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

Lärare besöker salen: c:a kl 15:00 och 16:30

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

Formalia: I programhuvudet på respektive m-fil ska du skriva in din dators nummer (=tentamenskod). Du hittar numret i gul text i nedre högra hörnet på skrivbordet (skärmbakgrunden).

Programmen du skriver ska vara tydliga och enkla att följa med förklarande kommentarsatser. Programhuvud skall fyllas i, och speciellt funktionsfiler skall beskrivas så kommandot `help` ger en utomstående person anvisning om hur programmet skall anropas.

Du skall arbeta mot mappen C:\\_MATLAB\_TENTA\_ . Under tentamenstiden används endast denna mapp med undermappar. Programfilerna till respektive uppgift finns och skall efter bearbetning sparas i tillhörande mapp uppgift1-uppgift5. Du får bara lämna en lösning till varje uppgift. Flera alternativa lösningar ger direkt noll poäng på uppgiften. Däremot kan en delvis löst uppgift ge poäng.

När du är klar med tentamen, stäng av Matlab och logga ut från datorn. Tentamensten ska lämnas in.

Rättning: Resultatet anslås senast torsdagen den 9 september på avdelningen för dynamik, Inst för Tillämpad mekanik, Hörsalsvägen 7, plan 3 i M-huset. Granskning sker fredagen den 10:e och tisdagen den 14:e september kl 12.30-13.00.

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

Betyg U < 16p ; 16p ≤ Betyg 3 < 24p ; 24p ≤ Betyg 4 < 32p ; Betyg 5 ≥ 32p.

**LYCKA TILL!**

1. Modifiering av program. (10p)

- a) I mappen uppgift1 finns programmet `Uppgift1a.m` i form av en kommandofil (skript) som plockar ut de två största elementen för varje rad i matrisen  $\mathbf{A}$  och sparar dessa på motsvarande rad i matrisen  $\mathbf{B}$ . Effektivisera programmet genom att fördimensionera (förallokera)  $\mathbf{B}$ . (2p)
- b) I programmet `Uppgift1b.m` finns en while-loop. Skriv om programmet med en for-loop som ersätter while-loopen utan att beräkningsresultaten ändras. (2p)
- c) Funktionsfilen `Uppgift1c.m` sorterar elementen i en  $5 \times 3$ -matris i stigande ordning radvis där element (1,1) är det minsta elementet och det sista elementet (5,3) är det största elementet. *Modifera* funktionsfilen så att funktionen fungerar för matriser av godtycklig storlek. Funktionen skall ha samma in- och utvariabler som innan, och du skall inte konstruera en helt annan lösning genom att skriva om funktionsfilen från början. (2p)
- d) I programmet `Uppgift1d.m` skall följande två problem lösas: Beräkna roten  $x$  till ekvationen  $g(x) = -2$  samt numeriskt beräkna derivatan i punkten  $x = 3$ . Funktionen  $g(x)$  defineras som
- $$g(x) = 3 \sin(e^{-0.6x} + 1) + 0.5x^3$$
- Rätta till så att programmet fungerar som avsett (det finns två allvarliga fel). (2p)
- e) Programmet `Uppgift1e.m` är en kommandofil (skript). Gör programmet mer generellt genom att skriva om `Uppgift1e.m` till en funktionsfil. Välj lämpliga in- och utvariabler utifrån programmets funktion. (2p)

Om du vill börja om från början med uppgiften kan du kopiera innehållet från mappen `uppgift1_original` till mappen `uppgift1`.

## 2. Programmeringsuppgift (8p)

Utböjningen  $w$  hos en fritt upplagd balk med en punktlast av storleken  $P$  riktad nedåt beskrivs av uttrycket

$$w(x) = \begin{cases} -\frac{P(1-\alpha)x(L^2-(1-\alpha)^2L^2-x^2)}{6EI} & \text{för } x \leq \alpha L, \\ -\frac{P(1-\alpha)x(L^2-(1-\alpha)^2L^2-x^2)}{6EI} - \frac{P(x-\alpha L)^3}{6EI} & \text{för } x > \alpha L. \end{cases} \quad 0 \leq x \leq L$$

där  $L$  är balkens längd,  $EI$  är böjstyvheten och  $\alpha L$ ,  $0 < \alpha < 1$ , är avståndet från origo till där punktlasten angriper.

Skriv ett program `Uppgift2.m` som ritar funktionerna  $w(x)$  i för intervallet  $0 \leq x \leq L$ . Parametrarna  $P$ ,  $L$ ,  $\alpha$  och  $EI$  skall ges värden genom frågor i kommandofönstret. Funktionen  $w(x)$  skall ritas med svart heldragen linje med linjetjockleken 2. Dessutom skall en rät linje ritas mellan punktern  $(0,0)$  och  $(0,L)$ , denna linje skall vara grön streckad.

Den största utböjningen kan beräknas som  $w(\hat{x})$  för  $\hat{x} = L\sqrt{\frac{2\alpha-\alpha^2}{3}}$ . Detta värde skall skrivas ut som text i figurfönstret på formen:

Största utböjning är XX m

## 3. Programmeringsuppgift (8p)

I textfilen `Slumpsiffror.txt` finns heltal ordnade som en matris. Din uppgift är att skriva ett program för att analysera hur skillnaden mellan största och minsta tal är fördelad.

- Skriv en funktionsfil, `StorstaSkillnad.m`, som givet en radvektor med tal beräknar skillnaden mellan det största och det minsta talet i vektorn. (2p)
- Skriv ett huvudprogram, `Uppgift3.m`, som läser in filen `Slumpsiffror.txt`, och för varje rad i filen tar reda på skillnaden genom att anropa funktionsfilen `StorstaSkillnad.m` en gång för varje rad. Spara resultatet i en vektor. (3p)
- Antag att skillnaderna sparats i vektorn  $S$  (som har lika många element som det finns rader i `Slumpsiffror.txt`). Det så kallade *kumulativa* medelvärdet  $M_{kum}$  definieras som en vektor där element  $k$  beräknas som medelvärdet av elementen på position 1 till och med  $k$  ur vektorn  $S$ , dvs:

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

Beräkna och plotta det kumulativa medelvärdet som funktion av elementnummer. (3p)

#### 4. Programmeringsuppgift (8p)

I tärningsspelet Yatzy används 5st sexsidiga tärningar och det gäller att med upprepade kast uppnå olika tärningskombinationer för att samla poäng. Ett föreklat fall skall undersökas här.

Skriv ett program `Uppgift4.m` som simulerar *ett* kast med 5 tärningar genom att skapa en vektor med 5 st heltal, vardera uniformt slumpmässigt fördelat 1-6. Programmet skall vidare ta reda vilken som är den mest förekommande siffran och hur många av denna siffra som förekommer i vektorn. Vid fallet att två siffror finns lika många gånger (två par eller alla siffror olika) skall den högsta siffran räknas. Därefter skall resultatet skivas ut med formaterad utskrift på skärmen. Nedan följer två exempel på skärmutskrifter (från två olika körningar av programmet):

```
Kast: 5 3 5 5 1, Resultat: tre femmor
```

```
Kast: 6 6 1 4 1, Resultat: två sexor
```

Om alla siffror är lika skall även texten `YATZY!` skrivas ut på en ny rad. I fallet att resultatet är ett ental (alla siffror olika) skall ändelsen vara grammtiskt korrekt "-a" istället "-or".

#### 5. Programmeringsuppgift (6p)

Betrakta matrisen  $C$ :

$$C = \begin{bmatrix} 2 & -2 & 3 & 0 \\ 1 & -3 & 4 & 0 \\ 9 & 4 & 2 & 3 \end{bmatrix}$$

Skriv en funktionsfil, `Uppgift5.m` som skapar tre olika radvektorer genom att ta elementen i en given matris (av godtycklig storlek) i olika ordning. Matrisen  $C$  används bara för att illustrera i vilken ordning element skall tas, medan din lösning skall fungera för alla matriser oavsett storlek.

- Första vektorn skall bestå elementen radvis, i exemplet med  $C$  blir det: `[2, -2, 3, 0, 1, -3, 4, 0, 9, 4, 2, 3]` (1p)
- Andra vektorn skall bestå elementen kolumnvis räknade bakifrån, i exemplet med  $C$  blir det: `[3, 0, 0, 2, 4, 3, 4, -3, -2, 9, 1, 2]` (2p)
- Tredje vektorn skall bestå av elementen en diagonal i taget, från vänster till höger, med början uppe till höger. I exemplet med  $C$  blir det: `[0, 3, 0, -2, 4, 3, 2, -3, 2, 1, 4, 9]` (3p)  
Ledning: En ansats kan vara att först utarbeta en lösning för kvadratiska matriser.