

Tentamen för kursen TME135 Programmering i Matlab för M1

Tid: 19 augusti 2011 kl 8:30-12:30

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 9:30 och 11:00.

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-uppgift6. 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 onsdagen den 7 september på avdelningen för Dynamik, Inst för Tillämpad mekanik, Hörsalsvägen 7, plan 3. Det kan också ses i studentportalen några dagar senare. Granskning sker fredagen den 9:e och tisdagen den 13: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 ; $16p \leq$ Betyg 3 < 24p ; $24p \leq$ Betyg 4 < 32p ; Betyg 5 \geq 32p.

LYCKA TILL!

1. Småuppgifter (6p)

- a) I filen `Standard.m` skall följande uttryck beräknas för en vektor \mathbf{x} (med n st element)

$$s = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (x_k - \bar{x})^2}$$

där \bar{x} är medelvärdet av elementen i vektorn \mathbf{x} . Rätta till programkoden så att resultatet blir rätt (testa med `std` i Matlab). (2p)

- b) Lös det linjära ekvationssystemet

$$\begin{cases} 1z_2 + z_4 = 3 \\ \frac{1}{7}z_1 - 2 = -z_4 \\ 2z_1 = 2 - \frac{2}{3}z_2 \\ z_1 - z_2 - 3z_4 = z_3 - 3 \end{cases}$$

på formen $\mathbf{A}\mathbf{z} = \mathbf{b}$ genom att först skapa matrisen \mathbf{A} och vektorn \mathbf{b} och därefter beräkna \mathbf{z} . Spara dina kommandon i filen `Ekvsys.m`. (2p)

- c) En primitiv funktion $F(x)$ kan beräknas med hjälp av numerisk integration av funktionen f enligt uttrycket

$$F(x) = \int_0^x f(s) ds$$

Skriv funktionsfilen `Primitiv.m` så att den för givet skalärt värde x och funktionshandtag f beräknar $F(x)$. (2p)

2. Uppgift (6p)

En partikel färdas i en bana runt origo i ett 3D-koordinatsystem. Partikelns läge ges av kurvan $\{x(t), y(t), z(t)\}$, där koordinaterna är funktioner av tiden $0 \leq t \leq T$. I filen `PositionsData.mat` finns fyra vektorer, \mathbf{t} \mathbf{x} \mathbf{y} \mathbf{z} , lagrade så att $\mathbf{x}(\mathbf{k})$ motsvarar $x(t_k)$. Din uppgift är att skriva ett program `Partikelbana.m` som skapar två figurer enligt:

- a) Rita partikelkurvan med kommandot `plot3`. (2p)

- b) Beräkna avståndet till origo, $r(t)$, som en funktion av tiden med Pythagoras sats

$$r(t) = \sqrt{x(t)^2 + y(t)^2 + z(t)^2}$$

Plotta avståndet r som funktion av tiden i en ny figur. Rita också in en horisontell linje på y -koordinaten $0.75r_{\max}$ där r_{\max} är det största värdet på $r(t)$ under tidsintervallet. Den horisontella linjen skall sträcka sig över det tidsintervall som definieras av vektorn \mathbf{t} . (4p)

3. Uppgift (5p)

För en rätvinklig triangel gäller mellan längden på katerna k_1 och k_2 (sidorna som ansluter till den räta vinkeln) och hypotenusan h (sidan motstående den räta vinkeln) sambandet

$$k_1^2 + k_2^2 = h^2$$

Triangelns area A ges av

$$A = \frac{k_1 k_2}{2}$$

Skriv ett program `Triangel.m` där användaren kan ange längden på antingen: 1) en katet och hypotenusan, eller: 2) de två kateterna. Programmet får fråga användare vilket av fallen som är aktuellt. Programmet skall sedan beräkna längden på den återstående sidan samt triangelns area, och skriva ut resultatet på skärmen med formaterad utskrift. Utskriften skall göras med 2 decimaler, enligt:

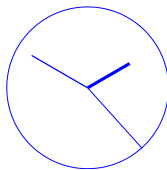
Den återstående kateten är 1.23 och arean är 45.67

Om två kateter anges skall utskriften vara på formen

Hypotenusan är 1.23 och arean är 45.67

4. Uppgift (9p)

Filen `KlockaAnalog.m` är ett huvudprogram som skall läsa in ett klockslag via fråga i kommandofönstret och sedan rita en analog klocka, enligt figur 1. Huvudprogrammet är färdigt, men använder sig av tre funktionsfiler som inte är skrivna. Din uppgift är att tolka huvudprogrammet och sedan skriva de tre funktionsfilerna enligt följande:



Figur 1: Exempel på klocka (kl 14:50:23)

- Funktionsfilen `RitaVisare.m` ritas en visare som en linje från origo, i viss vinkel från positiva x-axeln. Vinkel, längd och tjocklek ges som invariabler. Funktionen har ingen utvariabel. (3p)
- Funktionsfilen `TidRadianer.m` beräknar respektive visares vinkel (i radianer) utifrån klockslag. Invariabel är ett klockslag som anges som en vektor med tre element, t.ex `[14 50 23]`. Utvariabel är en vektor med vinkel för respektive visare. Timvisare och minutvisare visar bara hela timmar respektive hela minuter. (3p)
- Funktionsfilen `Texttolkare.m` tolkar en teckensträng för tiden på formen `TT:MM:SS` och skapar en vektor med tre element `[TT MM SS]`. (3p)

5. Uppgift (8p)

En matris \mathbf{A} kan användas för att beskriva en labyrint genom att varje element \mathbf{A}_{ij} antingen har värdet 0 (gång) eller 1 (vägg). I filerna `Labyrint_1.txt` och `Labyrint_2.txt` finns två labyrinter där det finns en sammanhängande väg av 0:or mellan vänster och höger sida. Med Matlabs funktion `spy` kan vi titta på matrisen \mathbf{A} och på så sätt se labyrinten.

För att lösa labyrinten genom att hitta den sammanhängande vägen från vänster till höger kan vi använda oss av algoritmen nedan. Strategin är att söka upp slutet på blindgångarna (0:or omgivna av tre stycken 1:or) och ersätta dem med 1:or. Detta upprepas tills inga fler blindgångar hittas, och det som återstår är (förhoppningsvis) endast den rätta lösningen.

- 1) Låt i löpa över rad 2 till näst sista raden i \mathbf{A}
- 2) Låt j löpa över kolumn 2 till näst sista kolumnen i \mathbf{A}
- 3) Om \mathbf{A}_{ij} är 0 och precis tre stycken av de angränsande elementen $\mathbf{A}_{i+1 j}$, $\mathbf{A}_{i-1 j}$, $\mathbf{A}_{i j+1}$, $\mathbf{A}_{i j-1}$ är 1 skall \mathbf{A}_{ij} få värdet 1
- 4) Upprepa 1-3 så länge minst ett element i \mathbf{A} ändras per cykel.

Skriv ett program som läser in en labyrint i definerad i en textfil liknande `Labyrint_1.txt`. Programmet skall därefter hitta lösningen med algoritmen ovan. Till sist skall både den ursprungliga och den lösta labyrinten ritas upp i två separata figurfönster som skall förses med lämpliga titlar. Spara programmet i filen `LabyrintLosare.m`.

6. Uppgift (6p)

I textfilen `TalSerie.txt` finns en lång lista med siffror 0-9. Skriv ett program `HittaSekvens.m` som hittar den längsta sekvens av likadana siffror i rad i listan. Programmet skall skriva ut vilken siffra det är, hur många siffror det finns i rad och på vilken plats i listan sekvensen startar.