Tentamen för kursen TME135 Programmering i Matlab för M1

Tid: 18 oktober 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 tisdagen den 8 november på avdelningen för Dynamik, Inst för Tillämpad mekanik, Hörsalsvägen 7, plan 3. Det kan också ses i studentportalen några dagar senare. Granskning sker onsdagen den 9:e och tisdagen den 15:e september kl 12.00-13.00.

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) Lös det linjära ekvationssystemet

$$\begin{cases} 2y_1 + 5y_3 = 1 - 7y_4 \\ 4y_2 + 3y_3 + 7y_4 - 3 = 0 \\ 4y_2 + \frac{1}{3}y_3 = 7y_1 + \frac{8}{3}y_4 \\ \frac{9}{5}y_3 = 2y_2 \end{cases}$$

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

   b) Funktionsfilen `AntiFiltrering.m` söker igenom alla element i en vektor, och ersätter tal större än 5 eller mindre än -5 med 0 och returnerar den modifierade vektorn. Ersätt for-loopen med en vektoriserad operation under villkor. Låt programkod du tar bort stå kvar som kommentarer. (2p)

   c) Några astronomiska data för de de fyra inre planeterna är angivna i nedanstående tabell:

Namn	Banellips	Massa
Merkurius	[0,47 0,31]	0,06
Venus	[0,73 0,72]	0,82
Jorden	[1,0 0,98]	1,00
Mars	[1,7 1,4]	0,11

   Skapa en struktur `planeter` som innehåller dessa data. Strukturen skall vara organiserad så att `planeter(1)` innehåller datan om Merkurius, `planeter(2)` om Venus osv. Fälten skall ha namn som rubrikerna i tabellen. Skriv tilldelningssatserna i filen `Planetdata.m`. (2p)

2. Uppgift (4p)

   Skriv ett program som använder `fzero` för att lösa ekvationen

$$g(x) = 3, \quad \text{med} \quad g(x) = \sqrt{|x|}\sin(2x) + \frac{3x}{2}$$

   Programmet skall rita upp funktionen $g(x)$ för intervallet $-5 \leq x \leq 5$ och förses med rutnät (grid). I figurfönstret skall ekvationens lösning skrivas ut som text på formen ”Lösningen är x=XXXX” på lämplig plats.

3. Uppgift (6p)

En andragradsekvation kan skivas på formen $ax^2 + bx + c = 0$. Lösningen ges av ("pq-formeln") som

$$x = -\frac{b}{2a} \pm \sqrt{D}, \quad \text{med} \quad D = \left(\frac{b}{2a}\right)^2 - \frac{c}{a}$$

Lösningen kan vara något av tre fall: två reella rötter ($D > 0$), en reell dubbelrot ($D = 0$) och två komplexa rötter ($D < 0$). Skriv ett program, `Andragradslosare.m`, som ber användaren om konstanterna a, b och c beräknar lösningen till andragradsekvationen och ger beroende på fall någon av följande utskrifter

```
Två reella rötter x1=XXXX och x2=YYYY
```

eller

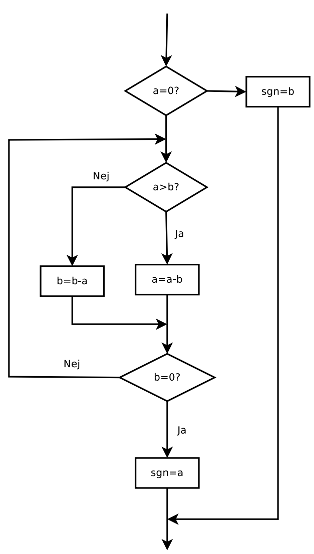
```
En dubbelrot x=XXXX
```

eller

```
Två komplexa rötter x1=XXXX och x2=YYYY
```

4. Uppgift (4p)

En variant av Euklides algoritm för att hitta största gemensamma nämnare mellan två tal ges av ett flödesschema Implementera denna algoritm som en funktionsfil i `StorstaGemenNamnare.m`.



Figur 1: Variant av Euklides algoritm

Till exempel skall anropet `StorstaGemenNamnare(44,33)` ge svaret 11.

5. Uppgift (10p)

Vårt solsystem består av solen samt åtta planeter (numera benämner pluto som en s.k. dvärgplanet). De färdas i banor som ligger i stort sett i ett plan, varför vi kan uttrycka deras massa samt position i ett vanligt koordinatsystem med solen i origo enligt följande tabell:

	namn	x (AU)	y (AU)	M (JM)
1	Solen	0	0	333 000
2	Merkurius	-0,23	-0,32	0,06
3	Venus	-0,29	-0,66	0,82
4	Jorden	0,95	0,31	1,00
5	Mars	-0,21	1,5	0,11
6	Jupiter	4,4	2,8	318
7	Saturnus	-9,0	-2,9	95
8	Uranus	20	1,0	15
9	Neptunus	27	-14	17

Tabellen finns i filen `Planetdata.txt`. Enheterna är 1 AU (astronomisk enhet, jordbanans medelradie) = 1,496$\times 10^{11}$ meter samt 1 JM (jordmassa) = 5,97 $\times 10^{24}$ km.

Enligt Newtons gravitationslag kan gravitationskraften F mellan två föremål med massorna M_i och M_j med avståndet r beräknas som

$$F = G\frac{M_i M_j}{r^2}m \quad \text{med} \quad r = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

där $G = 6,674 \times 10^{-11}$ N m^2/kg^2 är gravitationskonstanten.

a) Skriv en funktionsfil, `NewtonLag`, som beräknar gravitationskraften mellan två planeter givet deras position och massa. Om $r = 0$ skall F sättas till Not-a-Number (`NaN`). (3p)

b) Skriv ett huvudprogram, `Gravitationskrafter`, som läser in textfilen `Planetdata.txt`, och sedan beräknar kraften F för alla kombinationer av två himlakroppar. Krafterna skall sparas i en matris F_{alla}. (5p)

c) Utvidga programmet så att det tar reda på vilken som är den största kraften mellan två himlakroppar. Skriv ut värdet på skärmen samt numret på de himlakroppar den verkar mellan. (2p)

6. Uppgift (10p)

Som en del i uppskattning av en produkts livslängd görs ofta utmattningsberäkningar. Vid dessa beräkningar används ofta uppmätta data från fysiska experiment eller tester. Den resulterande mätdatan har ofta en hög grad av komplexitet, vilket det svårt att få en korrekt uppskattning av en produkts livslängd, därför försöker man att förenkla mätresultaten genom att identifiera s.k lastcykler. Denna uppgift går ut på att göra an sådan förenkling av mätdata som är sparat i filen `VibrationsData.mat`. I filen är tiden sparad som variabeln `t` och förskjutningen (förflyttning av en punkt) som funktion av tiden i variabeln `K`. Som ledning ges figuren nedan.

a) Ladda filen `VibrationsData.mat` som ger variablerna `K` och `t`. Skapa en figur som plottar `K` mot tiden `t`. Linjen skall vara av typen streckad blå. Text på x och y axlarna skall tydligt visa vad som plottas. Beräkna sedan medelvärdet K_{medel} av `K` och plotta detta som en punktad röd linje över mätdatan. Om du inte lyckas ladda filen kan datan skapas med Matlab-kommandot

```
t=linspace(0,25);
K=3.5+sin(t)+rand(1,length(t)).*sin(4*t)+0.25*rand(1,length(t)).*sin(6*t);
```

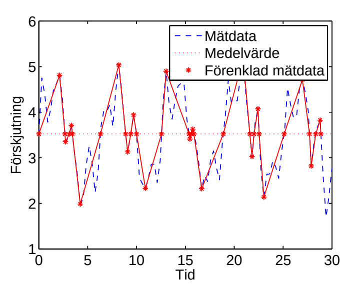
för att skapa `K` och `t`. Observera att om du inte laddar in `VibrationsData.mat` utan använder detta kommando, kommer det ge poängavdrag. (3p)

c) Finn de punkter där `K` korsar medelvärdet. En sådan korsning inträffar när två på varandra följande punkter K_i och K_{i+1} befinner sifg på varsin sida av K_{medel}. Markera dessa punkter med en röd stjärna i figuren. Tidpunkten t_{kors} då K skär K_{medel} mellan tiderna t_i och t_{i+1} beräknas enligt formeln

$$t_{\text{kors}} = \left(\frac{t_{i+1} - t_i}{K_{i+1} - K_i} \right) (K_{\text{medel}} - K_i) + t_i$$

där i är index.

d) Mellan två på varandra följande skärningspunkter skall extremvärdet hittas, dvs det högsta värdet i de områden där K är större än K_{medel} och det lägsta värdet i de områden där K är mindra än K_{medel}. Även dessa skall markeras med en röd stjärna. Slutligen skall heldragna röda linjer dras mellan stjärnorna samt en förklaringsruta som beskriver vad de olika linjerna visar skapas.



Figur 2: Exempel på figur. Textstorlek har modifierats för läsbarhet.