



KTH Datavetenskap
och kommunikation

Spektrala transformeringar

MATLAB-övning

1 Hjälpsystemet

Börja med att bekanta dig med matlabs inbyggda hjälpsystem, här kan man få information om inbyggda funktioner, nyckelord och koncept. Prova att skriva

```
help funktion
```

för några olika funktioner, t.ex. `diag`, `exp`, `magic` och `i`.

Du kommer att behöva kolla upp flera funktioner själv i uppgifterna nedan, men ofta får du tips om relevanta funktionsnamn.

Ibland vet man ju dock inte ens vad funktionen man söker heter, eller ens om den finns, då kan man använda

```
lookfor nyckelord.
```

Då letar matlab i alla hjälptexter efter nyckelordet. *OBS plågsamt långsamt!*

2 Matriser och arrayer

Gå igenom avsnitt 2 *arrays and matrices* i *Crash course in MATLAB*. Prova funktionerna vart eftersom du läser om dem. Gör sedan nedanstående övningar.

- skapa en s.k. magisk kvadrat med `a=magic(7)` och verifiera att summan i alla rader och kolumner är konstant (*tips: sum*).
- plocka ut valfri rad respektive kolumn ur `a` (*tips: använd "kolon-notation"*)
- skriv ett uttryck som returnerar en ny, mindre matris genom att ta varannan kolumn och varannan rad från `a`
- vänd matrisen runt den horisontella respektive den vertikala axeln! (*tips: kan ses som en variant på ovanstående. Alternativt: lookfor flip...*)
- vänd matrisen längs huvuddiagonalen.
- skapa en ny matris `b` av samma storlek som `a`, som där varje element är kvadraten av motsvarande element i `a`.
- skapa en normerad variant av `a` där alla elementen ligger i intervallet $[0, 1]$. undvik loopar! (*tips: min, max*)
- skapa en ny matris `c` av samma storlek som `a`, där varje element är 1 om motsvarande element i `a` är större än 25, annars 0. (*tips: relop*)
- skriv ett uttryck som plockar ut alla tal som är delbara med 3 ur en vektor eller matris (*tips: find, mod*)

- skriv om nedanstående kod så att loopar undviks:

```

a = [1 .5 .3 .4];
w = [1 2 3 4];
y = 0;
for i = 1:length(a)
    y = y + a(i) * w(i);
end

```

- generera en vektor x av längden 1000 med simulerade resultat från en sexsidig tärning. Verifiera sedan att tärningen är välbalanserad! (*tips: rand, round, find*)
- Generera en mängd punkter med $10 * \text{rand}(2, 10)$. Sortera sedan dessa efter deras avstånd till origo.

3 Plottar, signaler och bilder

Läs om grafikmöjligheterna i matlab under avsnitt 5 i *Crash course in MATLAB*. Här kommer vi att koncentrera oss på `plot()`.

3.1 Enkla plottar

OBS! n betecknar ett heltal i nedanstående uppgifter.

- plotta $y(n) = |n|$ när $-10 \leq n \leq 10$
- plotta "gauss-klockan"

$$y(n) = e^{-\left(\frac{n}{5}\right)^2}$$

med gröna cirklar när $-10 \leq n \leq 10$

- plotta $x_1(n) = \cos \omega_1 n$ när $\omega_1 = \frac{\pi}{100}$ och $0 \leq n \leq 1000$
- plotta $x_1(n)$ ovan tillsammans med

$$x_2(n) = \sin \omega_2 n + \sin(\omega_1 + \omega_2)n$$

när $\omega_2 = \frac{\pi}{10}$, $\omega_1 = \frac{\pi}{100}$ och $0 \leq n \leq 1000$.

3.2 Phasors och fyrkantvågor

En phasor (fasvektor) beskrivs av

$$x(n) = e^{j\omega n}$$

Låt $0 \leq n \leq 1000$ och bestäm ω så att phasorns vinkel gör ett helt varv över detta intervall.

- Plotta realdel och imaginärdel av $x(n)$ i samma graf.

Funktionen

$$x(n) = -\frac{2j}{\pi}(e^{j\omega n} - e^{-j\omega n}) - \frac{2j}{3\pi}(e^{j3\omega n} - e^{-j3\omega n}) - \frac{2j}{5\pi}(e^{j5\omega n} - e^{-j5\omega n})$$

beskriver de tre första termerna i fourier-serien för en fyrkantvåg.

- Plotta funktionen.
- Plotta term 1, 1+2 och 1+2+3 i samma graf, och notera att funktionen blir mer lik en fyrkantvåg ju fler termer man har.
- Om du vill: lägg till fler termer efter samma mönster.

3.3 Bilder

Bilder i representeras i matlab som matriser - varje pixel motsvarar ett matriselement. För att läsa in en bild i matlab använder man funktionen `imread()`. Den läser de flesta format och returnerar en matris av pixelvärden mellan 0 och 255. Det finns dock två problem man kan stöta på: vissa format returnerar matriser av heltalstyp och dessa måste konverteras till `double` för att fungera som vanliga matlab-matriser. `I=double(imread('minbild.gif'))` gör detta.

Färgbilder returneras som en $M \times N \times 3$ -matris. Dessa kan konverteras till en vanlig 2D-gråskalematrix genom att ta medelvärdet av de tre färgplanen. `mean(imread('minbild.gif'),3)` tar hand om detta.

Sammantaget innebär detta att följande rad troligtvis är bra om man vill kunna läsa in många olika typer av bildfiler:

```
I = mean(double(imread('minbild.gif')),3);
```

För att titta på en bild använder man kommandona `image` resp. `imagesc` - den förra förväntar sig pixelvärden mellan 0 och 255 medan den senare skalar om bildens gråskalevärden automatiskt, vilket ofta är praktiskt.

Du behöver även skriva `axis image` (säkerställer att höjd/bredd-förhållandet är bra) samt `colormap(gray)` (gör att matrisen tolkas som en gråskalebild).

Du ska nu pröva dessa funktioner och även behandla bilden på olika enkla sätt med hjälp av matlabs vanliga matrishantering. Du får använda valfri bildfil som inte är för stor (ca 500x500 pixlar kan vara lagom). Det finns en lämplig testbild åtkomlig från kurshemsidan. Nedanstående punkter kan ses som inspiration - du måste inte göra alla och du får gärna prova andra operationer på bildmatrisen.

- Läs in din i matlab som en gråskale-matris och visa den på skärmen.
- Invertera bilden, så att mörkt blir ljust och vice versa!
- Beskär bilden, så du bara tittar på ett område i mitten *tips: använd matlabs kolon-notation för matrisindexering*
- Vänd bilden upp-och-ner (*tips: går att göra på samma sätt som föregående uppgift!*)
- Skala ner bilden genom att plocka bort varannan rad och varannan kolumn! Hur blir resultatet? Vad är problemet med denna metod för bildskalning?
- Gör bilden brusig genom att lägga till en slumpmässmatris (med lämplig skalfaktor)

4 Funktioner

Läs om funktioner i avsnitt 3 i *Crash course in MATLAB*.

Funktioner är ett utmärkt sätt att skapa återanvändbara algoritmer, och bringa reda i sin matlab-kod. Gör det till en vana att använda funktioner med vettiga in- och utparametrar så kommer dina matlab-program bli så mycket lättare att förstå.

Nu ska du göra om några av uppgifterna ovan till funktioner. Glöm inte att skriva hjälp-text för funktionerna (de kommentar-rader som följer efter funktionshuvudet). Testa sedan att skriva `help din funktion!`

- I tredje uppgiften i avsnitt 2 fick du reducera en matris genom att ta varannan rad och varannan kolumn. Skriv en generell funktion för detta! Inparametrar bör vara en matris, samt två heltal som anger vertikal och horisontell reduktionsfaktor (dvs 2 ger varannan rad, 3 var tredje etc.). Funktionen ska returnera den reducerade matrisen.
- Skriv en funktion som beräknar fourier-serie approximationen av en fyrkantvåg, med godtyckligt många termer! Generalisera utifrån ekvationen i uppgift 3.2 eller titta i formelsamlingen på kurshemsidan. Inparametrar ska vara sekvenslängd N , grundfrekvens ω_0 , samt K - den högsta frekvens som får ingå uttryckt i multiplar av grundfrekvensen. Returvärdet ska vara den approximerade fyrkantvågen. Detta innebär att med inparametrarna $N = 1000, \omega_0 = 2\pi/1000$ samt $K = 5$ bör du erhålla samma resultat som i summan i uppgift 3.2.

Testa din funktion genom att plotta resultatet för olika värden på K !