

## Lab A1

I den här kursen ska vi lära oss att använda de viktigaste numeriska metoderna med hjälp av programmeringsspråket och programmet MATLAB. Under datorsalsövningarna förväntas ni arbeta självständigt med matlab i tvåmannagrupper (utom wiki-uppgiften som individuell) och genomföra uppgifterna som ni hittar dels i detta dokument (Lab A-1) och på kurshemsidan (Lab A-2 och Lab B).

Laborationsmomentet i kursen består av Lab A1, Lab A2 och Lab B. Om Lab A1 redovisas innan bonusdatum (angivna på hemsidan) får man bonuspoäng till tentamen. Motsvarande gäller för Lab A2.

Planera ert arbete för laborationerna väl i förväg. Ni kommer säkert inte hinna göra alla uppgifter under datorsalsövningarna så planera in självständigt arbete utanför schemalagd tid så att ni kan redovisa under en datorsalsövning.

**Redovisning:** Redovisning görs lämpligtvis löpande, dvs när ni är färdiga med en komponent, och samla godkändunderskrifter av kursens labassur. Ställ er i kön i <http://queue.csc.kth.se> när ni vill redovisa eller har frågor.

Lycka till!



o. (redovisas ej)

- (a) **Kursregistrering i RAPP.** Det är viktigt att du kursanmäler dig i Rapp, <http://rapp.csc.kth.se>. Logga in där med ditt KTH-id och klicka på aktivera. (I RAPP kan du se rapporterade lab-bonuspoäng etc. Kontrollera under kursens gång att dina resultat rapporteras in.)
- (b) **MÖ-uppgifterna.** För att bekanta dig med MATLAB föreslår vi att du gör "Inledande laboration : bekanta dig med MATLAB", som finns på kurshemsidan. Arbeta igenom så många som möjligt av MÖ-uppgifterna men prioritera uppgifterna 1-5 och 9-12. Du hittar MÖ-uppgifterna på kurshemsidan.
- (c) Läs igenom hela denna laborationsbeskrivning, **framförall instruktionerna om wiki-uppgiften i slutet**. Om du väljer att göra wiki-uppgiften, bör den göras parallellt med de andra uppgifterna.

## 1. Flyttalsaritmetik. Kör det här programmet i MATLAB:

```
kv=0:15; fv=[]; gv=[];
for j=1:length(kv)
    z=pi+10^kv(j)
    fv(j)=(z-10^kv(j))-pi;
    gv(j)=z-(10^kv(j)+pi);
end
figure(1); semilogy(kv,abs(fv-gv))
```

Obs: Det går att använda copy-paste från denna PDF-fil.

Vad skulle gv och fv vara i exakt aritmetik? Vad blir gv och varför? För vilka värden på  $k$  gör vi störst fel? Varför?

## 2. Det här programmet löser ett linjärt ekvationssystem, mäter beräkningstiden, och beräknar normen av residualvektorn.

```
n=50; A=randn(n,n); b=randn(n,1);
tic();
x=gausselim(A,b);
t=toc()
norm(A*x-b)
```

För att köra programmet måste ni ladda ned en (naiv) implementation av Gauss-eliminering, gausselim.m och spara i arbetskatalogen:  
<http://www.math.kth.se/~eliasj/gausselim.m>

- (a) Kör programmet ovan. Jämför det med MATLABs inbyggda metod för linjära ekvationssystem, den s.k. backslashoperatorn:  $A \setminus b$ , genom att ändra programmet. Vilket program är snabbast och genererar en lösning med minst fel?
- (b) Fyll i tabellen nedan med beräkningstider. När ni fördubblar  $n$ , med vilken faktor förväntas beräkningstiden öka enligt teorin? Stämmer det alltid med teorin? Varför inte?

	gausselim	backslash
$n = 25$		
$n = 50$		
$n = 100$		
$n = 200$		
$n = 400$		

## Redovisning uppg 1-2

Vid redovisning av dessa uppgifter var redo att svara på frågor om resultaten. När ni redovisar ska ni ha lätt tillgängligt på skärmen eller utskrivet:

- Figur från uppgift 1
- Ifylld tabell för uppgift 2

Uppgift 1-2 godkänd av assistent (signatur, datum): \_\_\_\_\_



3. Vi vill bestämma samtliga rötter till följande ekvation

$$x - 4 \sin 2x - 3 = 0$$

- Rita grafen för  $y(x) = x - 4 \sin 2x - 3$  med kommandot `plot`. Experimentera med olika intervall i  $x$ -led och olika tabellsteg så att figuren till slut har med alla nollställen till  $y(x)$ .
- Bestäm den största och den minsta roten till ekvationen  $x - 4 \sin 2x - 3 = 0$  med Newton-Raphsons metod. Redovisa startgissningar, antal iterationer och resultat för de två rötterna.
- Newton-Raphsons metod sägs ha kvadratisk konvergens. Förklara hur man kan avläsa detta ur resultatutskriften.
- Är Newton-Raphson idiotsäker eller spårar den ur någon gång? Prova t.ex. med startvärdet  $x_0 = 7$ . Förklara beteendet. Motivera med hjälp av en lämplig figur.
- Bestäm den största roten med sekantmetoden med  $x_0 = 4$  och  $x_1 = 5$ . Behöver vi fler eller färre steg med sekantmetoden än Newton-Raphsons metod, för att till exempel nå ett fel som är mindre än  $10^{-10}$ . Motivera.

---

*Redovisning uppg 3*

---

Vid redovisning av dessa uppgifter var redo att svara på frågor om resultaten. När ni redovisar ska ni ha lätt tillgängligt på skärmen eller utskrivet:

- Grafer med funktionen, och alla lösningar beräknade
- Antal steg Newtons metod behöver och sekantmetoden behöver

Uppgift 3 godkänd av assistent (signatur, datum): \_\_\_\_\_

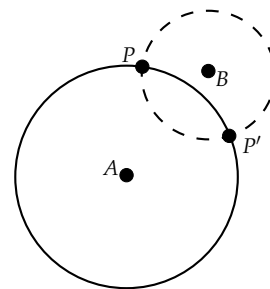


4. Koordinaterna till punkten  $P$  skall bestämmas genom att man mäter avstånden till två kända punkter  $A$  och  $B$ . Se figur till höger (Detta sätt att bestämma en punkts okända koordinater kallas inom geodesin för inbindningsmetoden och är det som satellitnavigeringssystemet GPS utnyttjar.) Då gäller

$$\begin{aligned}(x_A - x_P)^2 + (y_A - y_P)^2 &= L_A^2 \\ (x_B - x_P)^2 + (y_B - y_P)^2 &= L_B^2.\end{aligned}$$

Detta ekvationssystem har två lösningar, då det ju finns två punkter,  $P$  och  $P'$ , som båda ligger på detta avstånd till punkterna  $A$  och  $B$ . Antag att man vet att de kända punkternas koordinater är  $A = (93, 63)$  och  $B = (6, 16)$ , och de uppmätta avstånden är ungefär  $L_A = 55.1$  och  $L_B = 46.2$ . Ekvationssystemet består av två ekvationer med två obekanta. Lös detta olinjära ekvationssystem med Newtons metod i flera variabler. Hitta båda lösningarna.

För att rita cirklar i MATLAB kan ni göra såhär:  
`tv=0:0.01:2*pi`  
`a=1; b=3; r=2.2`  
`plot(a+r*cos(tv), b+r*sin(tv))`




---

*Redovisning uppg 4*

---

Uppgift 4 godkänd av assistent (signatur, datum): \_\_\_\_\_



5. **Sifferigenkänning.** Ladda ned filerna `minidigits.mat` och `plotdigit.m` och kör `load minidigits.mat` så att ni får tillgång till data som motsvarar handskrivna siffror. Vi ska klassificera siffrorna som sparas som kolumner i matrisen `testdata`.

<https://www.math.kth.se/na/SF1547/numd16/minidigits/>

Genom att analysera stora mängder data har vi förberett så kallade centroider som sparas som kolumner i matrisen  $C$ . Centroider är bilder som motsvarar i princip ett genomsnitt av många handskrivna siffror, i detta fall för siffran 2. Vi ska nu försöka automatiskt känna igen alla tvåor i vår `testdata`. Eftersom alla centroider i  $C$  ser ut som en tvåa, kommer även en summa av centroider se ut som en tvåa. För att testa om en siffra (dvs en kolumn i matrisen `testdata`) är en tvåa försöker vi hitta den bästa linjärkombinationen av centroider som passar till `testdata`-vektorn. Kom ihåg att en linjärkombination skrivs som (med  $C = [c_1, \dots, c_k]$ )

$$x_1 c_1 + \dots + x_k c_k.$$

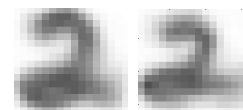
- Formulera problemet att hitta den bästa linjärkombinationen (dvs  $x_1, \dots, x_k$ ) som ett överbestämt linjärt ekvationssystem. Hur många okända variabler har vi? Vad är högerledet? Här krävs ingen programmering.
- Skriv ett program som för varje kolumn i `testdata` beräknar den bästa linjärkombinationen och sedan beräknar hur bra linjärkombinationen passar till testbilden, genom att beräkna  $\text{nv}(j) = \text{norm}(C \cdot x - \text{testdata}(:, j))$ , dvs normen av residualen av anpassningen.
- Ni har nu en vektor  $\text{nv}$  som är ett mått på hur bra linjärkombinationen passar till testsiffran. Om  $\text{nv}(j)$  är litet så borde `testdata(:, j)` se ut som en tvåa. Vi säger nu att om  $\text{nv}(j) < p$  för ett fixerat värde  $p$  klassificeras siffran som en tvåa. Välj  $p$  som medelvärde mellan  $\text{mean}(\text{nv})$  och  $\text{min}(\text{nv})$ . Den korrekta klassificering av siffran `testdata(:, j)` lagras i `testdatad(j)`. Analysera hur bra er klassificeringsmetod fungerar. Hur många procent av (alla) siffror klassificeras felaktigt som en tvåa? Hur många procent tvåor missar ni?

(För nyfikna: Kan ni hitta ett bättre värde  $p$ ?)

När ni laddat filen i MATLAB får ni en matris `testdata` där varje kolumn är en vektor av längd  $16^2 = 256$  och motsvarar pixelvärden i en svartvit bild med  $16 \times 16$  pixlar. Vi kan plotta bilden med funktionen `plotdigit`. Till exempel blir bild nummer 400 med `plotdigit(testdata(:, 400))` så här.



Så här ser de första två centroiderna  $C(:, 1)$  och  $C(:, 2)$  ut



### Redovisning uppg 5

Vid redovisning var redo att förklara:

- Hur ni löst det linjära överbestämda ekvationssystemet.
- Förklara vad ni gjort.
- Procentsatser för hur bra det blev.

Uppgift 5 godkänd av assistent (signatur, datum): \_\_\_\_\_



6. **Wiki-arbete.** Denna uppgift är inte obligatorisk, och man har rätt till bonus även om man inte har gjort denna uppgift. Har du genomfört denna uppgift vid labredovisningstillfället innan bonusdeadline ändras betygsgränser för betyg A och B (enligt hemsidan). Din övningsgrupp har i CANVAS en wiki-sida där ni ska lägga till och svara på uppgifter. Wiki-arbetet är individuellt och du behöver

- formulera minst 2 uppgifter (per person) från två olika block, och
- lösa minst 2 uppgifter (per person) från två olika block. Lös inte dina egna uppgifter.

Innan du redovisar skriv här vilka uppgifter du har löst (på formen GG-BB-UU, grupp, block, uppgiftsnummer)

\_\_\_\_\_ - \_\_\_\_\_ - \_\_\_\_\_ och \_\_\_\_\_ - \_\_\_\_\_ - \_\_\_\_\_

och vilka uppgifter du formulerat

\_\_\_\_\_ - \_\_\_\_\_ - \_\_\_\_\_ och \_\_\_\_\_ - \_\_\_\_\_ - \_\_\_\_\_

När laborationsassistenten skriver under "Hela lab godkänd" påminn om att hen ska anteckna era namn/personnummer och om ni gjort wiki-delen så att allt kan rapporteras in i systemet. Glöm ej att skriva erat namn högst upp på detta papper. Detta papper är eran dokumentation på genomförd lab. Spara det på lämpligt ställe tills ni ser resultatet i rapp.

Hela Lab A1 godkänd av  
assistent (signatur, datum): \_\_\_\_\_