

MAN 100 - MATEMATIK MED TILLÄMPNINGAR 1, 2003

INLÄMNINGSUPPGIFT NR 5: NUMERISK ANALYS och MATLAB

Uppgiften är värd 2 kurspoäng.

Inlämning till Yosief senast 5 januari 2004.

Uppgift nr 1 - Bästa steglängd vid differensapproximation

Vi ska studera följande så kallade differensapproximation av en derivata:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \quad (1)$$

där h är ett litet tal. Att högerledet går mot derivatan när h går mot 0 vet vi av derivatans definition. Genom Taylorutveckling, se kapitel 9 i PB, kan vi skriva

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2}h^2 + \frac{f^{(3)}(x)}{3!}h^3 + \dots$$

a) Visa att om f -värdena är exakta och h är litet så kan felet R_T i approximationen (1) (approximativt) uppskattas genom

$$|R_T| \leq \frac{Mh}{2}, \text{ där } |f''(x)| \leq M.$$

b) Anta nu att f -värdena inte är exakta men kan bestämmas med relativt fel begränsat av ϵ . Visa, under förutsättning att h är litet, att felet R_f i differensapproximationen (1) som beror på dessa fel i f -värdena kan uppskattas med

$$|R_f| \leq \frac{2\epsilon |f(x)|}{h}.$$

c) Totala felet i (1) kan alltså uppskattas med $\frac{Mh}{2} + \frac{2\epsilon|f(x)|}{h}$. Visa att detta fel minimeras då $h = 2\sqrt{\frac{\epsilon|f(x)|}{M}}$.

d) Låt nu $f(x) = \sin(x)$ och $x = 1$. Skriv ett MATLAB-program som approximerar derivatan $f'(1)$ med formeln (1) och steglängd $h = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots$. Jämför approximationen med exakta värdet $\cos(1)$. Rita i en log-skala, dvs använd **loglog** i stället för **plot**, ut felet som funktion av h . Verifiera att bästa h ungefär överensstämmer med det teoretiskt uträknade i c)-uppgiften. Maskintalet ϵ får du genom variabeln **eps** i MATLAB.

Inlämning: Härledning av optimalt h , MATLAB-program och figur som visar optimalt h -värde.

Uppgift nr 2 - Runges fenomen vid interpolation samt spline-interpolation

Interpolation innebär att man anpassar ett polynom av visst gradtal till ett antal punkter i planet så att polynomet går exakt genom punkterna och används som approximation

mellan punkterna. Om antalet punkter är $n + 1$ så är gradtalet på polynomet $\leq n$. Interpolation med högt gradtal kan leda till dålig approximation mellan punkterna som vi ska se genom att studera den så kallade Runges funktion $R(x) = \frac{1}{1+25x^2}$ på intervallet $-1 \leq x \leq 1$. Splines är styckvisa polynom av lägre grad, som man kan använda till interpolation i givna punkter. Fullständigare teori för interpolation och splines finns i föreläsningssanteckningarna i numerisk analys men behövs inte för att utföra denna inlämningsuppgift.

a) Välj nio likformigt fördelade punkter mellan -1 och 1: $x_i = -1 + 0.25 i$, $i = 0, 1, \dots, 8$ och bestäm $y_i = R(x_i)$ där $R(x)$ är Runges funktion ovan. Rita ut punkterna (x_i, y_i) i MATLAB.

b) Anpassa ett polynom av grad åtta till punkterna med **polyfit**. Rita ut polynomet med hjälp av **polyval** och **plot**.

c) Anpassa en kubisk spline till punkterna i a)-uppgiften med **spline** i MATLAB och rita ut den anpassande splinen med **plot**.

d) Vilken av anpassningarna, polynom av grad åtta eller spline, stämmer bäst med den bakomliggande funktionen $R(x)$?

Inlämning: MATLAB-program, MATLAB-grafik och svar på frågan.

Uppgift nr 3 - Minsta kvadrat och numerisk instabilitet

En planetbana är elliptisk och kan i ett (x, y) -koordinatsystem representeras av ekvationen $ay^2 + bxy + cx + dy + e = x^2$

a) Följande tabell innehåller observationer av en planets position (vid olika tidpunkter):

x	1.02	0.95	0.87	0.77	0.67	0.56	0.44	0.30	0.16	0.01
y	0.39	0.32	0.27	0.22	0.18	0.15	0.13	0.12	0.13	0.15

Lös problemet att anpassa ellipsen till mätdata genom att lösa det överbestämda ekvationssystemet i minsta-kvadrat-menig med MATLAB.

b) Minsta-kvadratproblemet är nästan rangdefekt, dvs kolonnerna i matrisen är nästan linjärt beroende. Lösningen är alltså illa-konditionerad. Testa effekten av instabiliteten genom att störa tabellvärdena med likformigt fördelade slumpstal i intervallet $[-0.005, 0.005]$ och lösa problemet på nytt. Slumptalen får Du med **rand** i MATLAB. Rita ut ostörd och störd ellipsbana tillsammans med mätpunkterna. Hur yttrar sig instabiliteten?

Inlämning: Grafik, m-fil som gör beräkningarna samt svar på frågan.

Uppgift nr 4 - Bildkomprimering med singularvärdesuppdelning (SVD)

Man kan digitalisera en bild med MATLAB genom att dela in bilden i ett rutnät och ge varje ruta ett tal motsvarande en färgkod eller ett värde på en gråskala vid svart-vit bild.

På detta sätt kan en bild representeras av en matris och omvänt kan en matris representera en bild om matrisens element ligger i ett intervall som representerar en färgskala. En god upplösning kräver dessvärre mycket stora matriser med stora krav på lagringsutrymme och beräkningstider om man ska manipulera bilden på något sätt. Frågan är då om man kan finna en acceptabel approximation av bilden, som kräver väsentligt mindre utrymme. Detta kan göras på många sätt bland annat med singularvärdesuppdelning (SVD). Approximationen av bilden utgörs då av en så kallad trunkerad singularvärdesuppdelning. Vi ska begränsa oss till svart-vita bilder och använda ett färdigt MATLAB-program för trunkerad SVD. Teorin för SVD står kortfattat i Lay's lärobok och i föreläsningssanteckningar Numerisk Analys kapitel 5.

- a) Digitalisera en bild med MATLAB-funktionen **imread**, som kan ta emot bilder i jpg-format. I brist på annat kan du ta bilderna **SpinningJenny.jpg** eller **svamp.jpg** från `/users/mdstud/ivar`. För att få kompakt lagring ger **imread** matrisen i formatet **uint8** så för att kunna utföra beräkningar på matrisen måste du konvertera den till formatet **double**.
- b) Låt MATLAB göra en full singularvärdesuppdelning (SVD) av matrisen genom att anropa funktionen **svd**.
- c) Vi ska nu se hur många termer som behövs i SVD för att få en användbar bild. För detta tillhandahåller kursen en MATLAB-funktion **svdimage**, som hämtas från `/users/mdstud/ivar`. Studera denna funktion så att Du förstår hur den fungerar och hur den används. Anropa funktionen och ange hur många termer Du tycker Dig behöva för snygg bild.
- d) Uppskatta graden av komprimering, dvs hur många element som behöver lagras för den approximerade bilden jämfört med den ursprungliga.

Inlämning: En kommandofil i MATLAB, som löser uppgiften och svar på frågorna om antal termer och grad av komprimering.

Uppgift nr 5 - Punkter i rymden

Givet koordinater för tio punkter i rymden ska Du avgöra om punkterna ligger på en cirkel. Om så är fallet ska Du bestämma medelpunkt och radie hos cirkeln.

Uppgiften kan lösas på flera sätt bland annat genom analytisk geometri. Här ska Du använda **QR-faktorisering**, en mer flexibel teknik som skulle kunna fungera även i högre dimension. Om QR-faktorisering läser Du i Lay's lärobok eller i föreläsningssanteckningarna Numerisk Linjär Algebra. Hur QR-faktorisering lämpligen används för att lösa uppgiften får Du försöka klura ut själv men det är lämpligt att ställa sig några frågor av typen:

- Vad är en cirkel?
- Vad är ett plan? Vad är ett underrum? Är ett plan alltid ett underrum?
- Hur bör mitt koordinatsystem se ut för att jag ska kunna använda cirkelns ekvation? Vad har detta med QR-faktorisering att göra?
- Kan cirkelns ekvation skrivas om så att "minsta-kvadrat" kan användas?

Det är även mycket lämpligt att först ha löst MATLAB-övningarna 64 och 80.

a) Diskutera dig fram till en lämplig lösningsmetod som använder QR-faktorisering.

b) Skriv en kommandofil i MATLAB som löser uppgiften.

c) Testa din ide' och ditt program på de tre datamängderna **cirkeldata1**, **cirkeldata2** och **cirkeldata 3** som Du hämtar från `/users/mdstud/ivar/rymdpunkter.mat` och sedan tar in till MATLAB med kommandot **load**.

d) Ange för vart och ett av de tre fallen om punkterna ligger i ett plan och i så fall om de ligger på en cirkel. Ange i förekommande fall radie och medelpunkt.

Inlämning: Beskrivning av Din lösningsteknik inklusive MATLAB-program och svar på frågorna om datamängderna ligger i ett plan, representerar cirkel och i så fall värdena på radie och medelpunkt.