Teknillinen korkeakoulu Apiola
Matematiikka

Mat-1.414 Matematiikan peruskurssi V2 kevät 2001
`http://www.math.hut.fi/teaching/v/3/H/`

Laskuharjoitus 10 (viikko 14 , 4–6.4.01)

AV ja LV

Ohjeita lopussa (kääntöp.), lisäksi `../L/SVD.mws`. AV-harjoituksissa on "kotitehtävänä" vain 1. Kannattaa suunnitella muiden osalta ratkaisua käsin, jotta pääsee hyvin työhön kiinni. Pidetään siis muuten ohjattu tekemisharjoitus myös ke.

Tehtävissä mainittujen itse kehitettyjen ohjelmien (+ fsolven) lisäksi kannattaa kokeilla lopussa mainittuja `NonlinearProgramming`-funktioita.

Kaikkia samantapaisia tehtäviä ei ole tarkoitus jokaisen tehdä. Tehtävistä 2,3,4,6 riittää valita kaksi. (Saa tehdä sopivia variaatioita.)

1. [AV]
   Osoita ketjusääntöä käyttäen, että SD eli JL, eli gradienttimenetelmässä uusi etsintäsuunta (suunta, johon edetään) on kohtisuorassa edellistä suuntaa vastaan (edellyttäen, että $\nabla f \neq 0$ uudessa pisteessä).
2. JL-menetelmän minimointiaskel voidaan toteuttaa antamalla Lmax, eli maksimaalinen askelpituusarvio ja puolittamalla sitä riittävän monta kertaa, mikäli $\varphi_u(Lmax) > \varphi_u(0)$ ($\varphi_u(t) = f(x_n + t\mathbf{u})$). Olkoon α ensimmäinen, näistä, joka alittaa $\varphi(0)$:n. Approksimoidaan $\varphi(t)$:tä 2. asteen interpolaatiopolynomilla, joka asetetaan pisteisiin $t = 0, t = \alpha/2, t = \alpha$. Tämän minimipiste otetaan oikean minimin approksimaatioksi.
   Näin on toteutettu SDstep työarkilla `..L/newtonsys.mws`. Tee itsellesi selväksi, testaa ja lisää viivahakukuvan piirto. Modifioi funktio mieleiseksesi ja mielellään niin, että käytetään LinearAlgebra-tyyliä. Huom! Kannattaa tehdä ensin komento komennolta ja vasta sitten kirjoittaa proc:ksi. (Jos et halua "ohjelmoida", voit tyytyä "skriptiin".)
   Käy läpi BF-kirjan Example 1 ss. 572–573 kynä/paperi- ja Maple-yhdistelmällä, askel kerrallaan ja kuvilla havainnollistaen. (Reitin voi piirtää myös 3D, onhan spacecurve keksitty.)
3. Kirjoita Maple-funktiota `fsolve` hyödyntävä mini-/maksimointifunktio vaikkapa `fmin` (tarkista nimi). Sisällytä Hessen matriisin posdef/negdef/indef-tarkistus.
   Selvitä funktiosi avulla harj9lv-esimerkkifunktion
   $f(x, y) = e^x(4x^2+2y^2+4xy+2y+1)$ minimit ja maksimit (ja mahd. satulapisteet) $\mathbb{R}^2$:ssa.
   Saat tietysti testata toisaalta myös omia funktioitasi sekä `NonlinearProgramming`-funktioita.
4. Minimoi funktio $f(x_1, x_2, x_3, x_4, x_5) = 0.5(x_1^4 - 16x_1^2 + 5x_1) + 0.5(x_2^4 - 16x_2^2 + 5x_2) + (x_3 - 1)^2 + (x_4 - 1)^2 + (x_5 - 1)^2$.
   Käytä yhdistelmää JL + Newton. Kokeile ainakin alkupistettä $(1, 2, 0, 2, 3)$
5. Olkoon annettu nelikulmio, jonka kärjet ovat pisteissä $p_1 = (0, 0)$, $p_2 = (1.8, 0)$,$p_3 = (1.5, 1)$,$p_4 = (0.3, 1.6)$. Määritä sisäpisteet $q_1 = (x_1, x_2)$ ja $q_2 = (x_3, x_4)$ siten, että pisteistä q_1 ja q_2 kaikkiin kärkiin p_i ja toisiinsa olevien etäisyyksien neliösumma on mahdollisimman pieni. (Olkoon tämä vaikka pieni informaatioverkko.)
6. Sovella SD/Newton-yhdistelmää (tms.) Harj. 8lv teht 4:ään (Suvin esittämään). Havainnollista tyylikkäin reittipiirroksin. Tämähän lienee aika hyvä testi ohjelmaparoille.
7. Ratkaise harj.9 LV LAODE ss- 352 - 354 -tehtävä sekä normaaliyhtälötyylillä että SVD:llä. Kts. `..L/SVD.mws`.
8. Singulaariarvohajoitelman (SVD) soveltaminen kuvankäsittelyyn, katso lopussa olevaa ohjetta.
   (Yllättävänä) sovellutuksena kokeillaan kuvatiedon pakkaamista. Ota jokin jpeg-kuva, lue se imread-komennolla Matlabiin matriisiksi. Kokeile, eri singulaariarvojen määrällä, kuinka hyvin saat kuvan rekonstruioduksi. Arvioi muistitilan säästöä.
9. BF: s. 560 teht. 8 (jaetaan pruju) Savimaahan asetettu levy, joka kannattaa painavaa kuormaa.
10. Tasossa liikkuvan 2-osaisen robotin käsivarren OAB piste O on kiinnitetty origoon. Pisteessä A on nivel, jolla varren osat OA ja AB on kiinnitetty toisiinsa. Varren OA ja x-akselin välinen kulma olkoon α ja varren OA ja AB välinen kulma olkoon β.
    Olkoot varsien pituudet d_1 ja d_2. Lausu pisteen B koordinaatit ohjauskulmien α ja β avulla.
    Robottia ohjattaessa halutaan annettua pistettä vastaavat ohjauskulmien arvot, ts. edellä lasketun funktion käänteisfunktion arvot.
    Olkoon $d_1 = 5$ ja $d_2 = 6$. Määritä Newtonin menetelmällä ohjauskulmat, joilla päästään pisteeseen $(10, 4)$, kun alkuarvoiksi otetaan $\alpha = 0.7$ $\beta = 0.7$ Onko ratkaisu 1-käsitteinen.
    Tee Newtonin iteraatiosta animaatio, joka näyttää käsivarren asennon kullakin iteraatioaskeleella. Samantien voit vaihdella kohdepistettä ja alkuarvoja. Mikä on käänteisfunktion määrittelyjoukko (robotin toiminta-alue)?
11. Epälineaarinen PNS. NMS-kirja S. 373 Exa 9.6, teht. p9-8 luminosity s. 380 Prujut jaetaan (Kirja: Kahaner–Moler–Nash: Numerical Methods and Software)

Ohjeita

0.0.1 Ammattimainen Maple-kirjasto optimointiin

Sain tänään (ma 2.4.) vastauksen kyselyyni Maple user-listalle ja asensin heti. Käyttöä varten tulee antaa komento

```
libname:= /p/edu/mat-1.414/maple/,libname; (kannattaa sijoittaa
                                            .mapleinit-tiedostoon.)
Sitten
with(linalg): with(LinearAlgebra): with(NonlinearProgramming);
```

Kts myös:
`http://www.mapleapps.com/powertools/optimization/html/nlpdemo.html`

0.0.2 SVD ja kuvatiedon pakkaaminen

Tässä on samalla tilaisuus kokeilla Matlabin kykyjä. Esimerkkinä käytän vaimoani esittävää kuvaa. Kannttaa kokeilla mieluummin suht. pienellä, minun kuvani on kokoa "medium", josta tulee 768 x 512 -kokoinen matriisi. (Kaikki muu on nopeata, mutta SVD kestää jonkin aikaa.)

`use matlab` ja `matlab`

Muista, että Matlabissa **tulostus estyy puolipisteellä**. Kärsimys voi näissä hommissa olla mittaamaton, jos unohdat tämän. (No, CTR-C auttaa).

```
X=imread('10a.jpg');   % Luetaan kuva Matlabiin.
X=double(X);           % muunnetaan liukulukumatriisiksi.
X=X(:,:,1)+X(:,:,2)+X(:,:,3); % Kuvamatriisi on 3-ulotteinen,lasketaan
X=X/26;        % sopiva kombinaatio tasoista (silkkaa kokeilua, tulee
               % kovasti sinisävyinen, mutta muuten selkeä).
image(X)       % Kuva näyttää nyt tältä.
size(X)
ans =
   768   512
X(1:10,1:10)  % Katsotaan 10x10-ylänurkkaa.

[U,S,V]=svd(X);  % Tehdään SVD - kestää ...

clf              % Clear graphics
                 % Leikkaa/liimaa ja painele ENTER:iä, kuvateksti
                 % kertoo käytettyjen singulaariarvojen lukumäärän.
p=30;
M=zeros(size(X));
for i=1:p
  M=M+S(i,i)*U(:,i)*V(:,i)';   % Tässä ne ulkotulot
  %figure(i)
  image(M);
  title(['Ensimmäiset ',num2str(i),' singulaariarvoa'])
  shg;pause
end;
          % Vaimoni tunnistan 16:n suurimman singulaariarvon perusteella!
          % (sigulaariarvoja yht. 512 kpl.)

figure(2)  % Toiseen ikkunaan vertailuksi
image(X)   % alkuperäinen

whos       % Muuttujat ja niiden muistintarve.
```

Jos et löydä kuvaa omasta takaa, voit käyttää Matlabin "oletuskuvaa", sinisävyistytä pikkupoikaa.

```
image
[X,Xm] = getframe;
X=flipud(X);   % Alkup on ylösalaisin, käännetään.
```

Sitten tehdään samat temput.