MAT 1120: Obligatorisk oppgave 2, H-09

Innlevering: *Senest fredag 30. oktober, 2009, kl.14.30, på Ekspedisjonskontoret til Matematisk institutt (7. etasje NHA). Du kan skrive for hånd eller med datamaskin, men besvarelsen skal uansett leveres på papir. Erfaringsmessig blir det lange køer både ved skriverne og utenfor ekspedisjonskontoret rett før innleveringsfristen, så det er smart å levere tidligere.* **Obligen skal leveres med en egen forside som du finner på**

http://www.uio.no/studier/emner/matnat/math/MAT1100/h07/obliger.xml

(det vil også være papirkopier av forsiden tilgjengelig ved innlevering). På samme side finner du regelverket for obliger ved Matematisk institutt. Husk spesielt å søke om utsettelse til studieinfo@math.uio.no før innleveringsfristen dersom du blir syk!

Instruksjoner: *Oppgaven er obligatorisk, og studenter som ikke får besvarelsen godkjent, vil ikke få adgang til avsluttende eksamen. For å få besvarelsen godkjent, må man ha minst 60% score (vekten til hver oppgave står på), og det vil bli lagt vekt på at man har en klar og ryddig besvarelse med gode begrunnelser. Alle svar skal begrunnes. Du kan få poeng på en oppgave selv om du ikke er kommet frem til et svar, og det er derfor viktig at du leverer inn alt det du har kommet frem til. Studenter som ikke får sin opprinnelige besvarelse godkjent, men som gjennom besvarelsen viser at de har gjort et reelt forsøk på å løse oppgavene, vil få én mulighet til å levere en revidert besvarelse.*

Det er lov å samarbeide og å bruke alle slags hjelpemidler. Den innleverte besvarelsen (tekst og programmer) skal imidlertid være skrevet av deg og gjenspeile din forståelse av stoffet. Er vi i tvil om at du virkelig har forstått det du har levert inn, kan vi be deg om en muntlig redegjørelse.

Obligatorisk oppgave i MAT1120 - Jacobis metode

Øyvind Ryan (oyvindry@ifi.uio.no)

10. oktober 2009

Egenverdiproblemer er svært viktige og fremkommer i de utroligste sammenhenger. Imidlertid innehar de fleste ingen analytiske løsninger og man må derfor ta i bruk numeriske metoder. Til nå har vi brukt MATLAB-funksjonen `eig()` eller PYTHON-funksjonen `linalg.eig()` når vi skal løse slike likninger numerisk, og de har nok for mange fremstått som mystiske "svarte bokser". I denne obligatoriske oppgaven skal du programmere din egen funksjon (i MATLAB eller PYTHON) som finner egenverdier numerisk ved bruk av av en mye brukt metode som kalles *Jacobis metode*. I Jacobis metode transformeres matrisen flere ganger ved såkalte similaritetstransformasjoner (Seksjon 5.2 i læreboka). Med andre ord, vi lager inverterbare matriser $P_1, P_2, ...$, og regner suksessivt ut

$$\begin{aligned}
A_1 &= A \\
A_2 &= P_1^{-1} A_1 P_1 \\
A_3 &= P_2^{-1} A_2 P_2 \\
\vdots \quad & \quad \vdots \\
A_{k+1} &= P_k^{-1} A_k P_k.
\end{aligned}$$

Oppgave 1

Vis at matrisene $A_1, A_2, A_3, ...$ alle har de samme egenverdiene. Skriv opp et uttrykk for egenvektorene til A_{k+1}, ved hjelp av egenvektorene til A_k, og bruk dette til å skrive opp et uttrykk for egenvektorene til A_k ved hjelp av matrisene $P_1, ..., P_{k-1}$, og egenvektorene til $A_1 = A$.

Idéen i Jacobis metode er å lage similaritetstransformasjonene slik at matrisen A_k blir mer og mer diagonal. Da vil nemlig det som står igjen på diagonalen være gode estimater på egenverdiene, på grunn av Oppgave 1, og siden egenverdiene står på diagonalen i en diagonalmatrise. Matrisene P_k vil ha følgende form:

$$P_k = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 0 \\ 0 & 1 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cos\theta & \cdots & \sin\theta & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & -\sin\theta & \cdots & \cos\theta & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 1 \end{bmatrix}, \tag{1}$$

der $(P_k)_{rr} = (P_k)_{ss} = \cos\theta$, $(P_k)_{rs} = -(P_k)_{sr} = \sin\theta$, $(P_k)_{ii} = 1$ (for $i \neq r, s$) og 0 ellers. Vi minner om at en kvadratisk matrise P kalles ortogonal hvis den er inverterbar, og $P^{-1} = P^T$ (Seksjon 6.2 i læreboka).

Oppgave 2

Vis at matrisene P_k definert ved (1) er ortogonale, slik at

$$(P_k)^{-1} = (P_k)^T = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 0 \\ 0 & 1 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cos\theta & \cdots & -\sin\theta & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \sin\theta & \cdots & \cos\theta & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 1 \end{bmatrix}.$$

Oppgave 3

Hvorfor kalles ofte P_k for en rotasjonsmatrise?

Oppgave 4

Lag en funksjon

```
function P=transmatr(r,s,theta,n)
```

som tar posisjonen (r, s), vinkelen θ, og dimensjonen n som argumenter, og returnerer matrisen (1).

Vinkelen θ i (1) skal for hver k velges slik at $A_{k+1} = P_k^{-1} A_k P_k$ blir "mer diagonal" enn A_k. Algoritmen for Jacobis metode består i at vi etter hver transformasjon $A_{k+1} = P_k^{-1} A_k P_k$ regner ut

$$\|A\|_F = \sqrt{\sum_{i \neq j}^{n} A_{ij}^2}, \tag{2}$$

og dersom denne verdien er mindre enn en bestemt toleranse, betraktes den resulterende matrisen som "diagonal nok". Du skal nå gjennom flere delsteg programmere en funksjon som utfører Jacobis metode.

Oppgave 5

Lag en funksjon

```
function fbo=frobnormoff(A)
```

som tar en reell $n \times n$-matrise A som parameter og returnerer $\|A\|_F$.

Oppgave 6

Lag en funksjon

```
function [r,s]=offabsmaks(A)
```

som tar en reell $n \times n$-matrise som parameter og returnerer posisjonen (r, s) til det elementet utenfor diagonalen med størst absoluttverdi.

Oppgave 7

Sett $B = P^T AP$, der P er rotasjonsmatrisen med vinkel θ og posisjon r og s. Vis at B er symmetrisk når A er det, og at vi da også har at

$$B_{rs} = B_{sr} = \sin\theta\cos\theta(A_{rr} - A_{ss}) + (\cos^2\theta - \sin^2\theta)A_{rs}. \tag{3}$$

Oppgave 8

Ved å regne ut B_{ij} som over, for alle andre i og j, kan det vises at

$$\sum_{\substack{i \neq j \\ (i,j) \neq (r,s) \\ (i,j) \neq (s,r)}} (A_{ij})^2 = \sum_{\substack{i \neq j \\ (i,j) \neq (r,s) \\ (i,j) \neq (s,r)}} (B_{ij})^2$$

(de ekstra interesserte kan forsøke å vise dette. Det er ikke vanskelig, men krever en del regning og forenklinger som vi dropper her). Forklar ut fra dette at, når A er symmetrisk, så vil r, s, θ gi en B med minst mulig $\|B\|_F$ når

1. r, s er valgt slik at $|A_{rs}|$ blir størst mulig,
2. $$\sin\theta\cos\theta(A_{rr} - A_{ss}) + (\cos^2\theta - \sin^2\theta)A_{rs} = 0 \tag{4}$$

Oppgave 9

Vis at hvis θ løser (4) så vil

$$\tan(2\theta) = \frac{2A_{rs}}{A_{ss} - A_{rr}}. \tag{5}$$

Oppgave 10

Du skal nå studere følgende kode

```
function [P,D]=jacobi(A)
   toleranse=10^(-4);
   n=size(A,1);
   P=eye(n);
   while frobnormoff(A)>toleranse
      [r,s]=offabsmaks(A);
      theta=atan( 2*A(r,s)/(A(s,s)-A(r,r)) )/2;
      Ptheta=transmatr(r,s,theta,n);
      A=inv(Ptheta)*A*Ptheta;
      P=P*Ptheta;
   end
   D=A;
```

Forklar hver del av denne koden, og hvilke deloppgaver over som blir brukt. Hva vil koden returnere?