UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i	MoD200 —
Eksamensdag:	15. desember 2003
Tid for eksamen:	14.30 – 17.30
Oppgavesettet er på 5 sider.	
Vedlegg:	Løsningsforslag
Tillatte hjelpemidler:	Ingen

Kontroller at oppgavesettet er komplett før
du begynner å besvare spørsmålene.

Oppgavesettet består av 6 deloppgaver med tilnærmet samme vekt.

Oppgave 1

Avgjør om følgende påstand er sann eller gal, begrunn svaret.

1a

Singulærverdiene til matrisen

$$A = \begin{bmatrix} -1 & -1 \\ 0 & -1 \end{bmatrix}$$

er alle positive?

Løsning Riktig. Siden matrisen har full rang er singulærverdiene positive. Vi kan også regne ut singulærverdiene direkte. Vi får

$$A^T A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

som har egenverdier $(3 + \sqrt{5})/2$ og $(3 - \sqrt{5})/2$. Den positive kvadratroten av disse tallene gir singulærverdiene. Vi finner $\sigma_1 = (1 + \sqrt{5})/2$ og $\sigma_2 = (\sqrt{5} - 1)/2$.

Oppgave 2 En ulikhet

La R være en 2×2 matrise på formen

$$R = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

(Fortsettes på side 2.)

hvor $c, s \in \mathbb{R}$ med $c^2 + s^2 = 1$. Vis at $\text{cond}_p(R) := \|R\|_p \|R^{-1}\|_p \leq 2$ for $p = 1, \infty$ og at vi har likhet for et passelig valg av c og s.

Løsning Siden $\|A\|_1 = \max_j \sum_i |a_{i,j}|$ finner vi $\|R\|_1 = |c| + |s|$. Ved Cauchy-Schwarz' ulikhet har vi $|c| + |s| \leq (|c|^2 + |s|^2)^{1/2}(1+1)^{1/2} = \sqrt{2}$. Vi har liket for $c = s = 1/\sqrt{2}$. Siden $R^{-1} = R^T$ får vi samme skarpe øvre grense for $\|R^{-1}\|_1$ og resultatet følger for 1-normen. Siden $\text{cond}_\infty(R) = \text{cond}_1(R)$ holder resultatet også for $p = \infty$.

Oppgave 3 Lineært ligningssystem

I denne oppgaven skal vi bruke plane rotasjoner til å løse et lineært ligningssystem $Ax = b$ som er nesten øvre triangulært. Vi antar at alle elementene under diagonalen er null untatt de i siste rad som alle kan være ulik null. Vi antar også for et positivt heltall n at $A \in \mathbb{R}^{n,n}$ er ikke-singulær og $b \in \mathbb{R}^n$. Som illustrasjon, for $n = 5$ har A formen

$$\begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ x & x & x & x & x \end{bmatrix}$$

Vi skal redusere A til øvre triangulær form ved en serie av plane rotasjoner på formen

$$R = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

hvor $c, s \in \mathbb{R}$ med $c^2 + s^2 = 1$.

3a

Skrive en Matlab `function [c,s]=rot(a,b)` som til gitte tall $a, b \in \mathbb{R}$ finner en plan rotasjon slik at

$$\begin{bmatrix} r \\ 0 \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \tag{1}$$

hvor $r = \sqrt{a^2 + b^2}$ bare beregnes i programmet. Dersom $a = b = 0$ setter vi $c = 1$ og $s = 0$. r skal beregnes på en spesiell måte. Vi setter $r = t\sqrt{(a/t)^2 + (b/t)^2}$ hvor $t = |a| + |b|$. Forklar hvorfor denne måten å beregne r på kan være gunstig for å unngå underflow og overflow.

Løsning Vi har

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} * \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} ca + sb \\ cb - sa \end{bmatrix}.$$

Dersom $(a, b) \neq 0$ kan vi sette

$$c = \frac{a}{r} \quad \text{og} \quad s = \frac{b}{r} \quad \text{hvor} \quad r = \sqrt{a^2 + b^2}.$$

(Fortsettes på side 3.)

Da finner vi at (1) holder. Hvis $b = 0$ og $a \neq 0$ er $c = 1$ og $s = 0$ og vi bruker også disse verdiene for $a = b = 0$. Kradrering av f.eks. a kan føre til at a^2 blir så liten at den blir satt til null (underflow) eller blir så stor at vi får overflow. Ved å beregne r på den spesielle måten unngår vi dette.

Matlabprogrammet kan være som følger:

```
 function [c,s]=rot(a,b) t=abs(a)+abs(b);

if b==0
    c=1; s=0; return;
end r=t*sqrt((a/t)^2+(b/t)^2); c=a/r; s=b/r;
```

Vi setter $A_1 = A$ og $A_{k+1} = R_k A_k$ for $k = 1, \ldots, n-1$ hvor R_k er en plan rotasjon som bruker posisjon (k, k) i A_k til å fremskaffe en null i posisjon (n, k) i A_{k+1} for $k = 1, \ldots, n-1$. R_k atskiller seg fra identitetsmatrisen kun i rad og kolonne k og n og har formen

$$R_k = \begin{bmatrix} 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & & \vdots & & & & \vdots \\ 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & c & 0 & \cdots & 0 & s \\ 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & \cdots & 0 & -s & 0 & \cdots & 0 & c \end{bmatrix}$$

3b

Forklar hvorfor A_n blir øvre triangulær. Skriv en Matlab `function x=rotsolve(A,b)` som beregner $U = A_n$ og $c = R_{n-1} \cdots R_1 b$ og så finner x ved tilbakesubstitutsjon. Vi lagrer alle A_k-ene i A etterhvert som de beregnes og vi antar at det er gitt en `function x=utrisolve(U,c)` som løser det øvre triangulære systemet $Ux = c$ ved tilbakesubstitusjon. Du skal ikke skrive innmaten i `utrisolve`.

Løsning Vi bruker induksjon på k for å vise at A_n er øvre triangulær. Anta at under diagonalen i A_k er det kun elementene $a_{n,k}, \ldots, a_{n,n}$ som kan være forskjellig fra null. Dette gjelder for $k = 1$. Anta det gjelder for en k. Matrisen $R_k A_k$ atskiller seg fra A_k kun i rad k og n og vi har

$$\begin{bmatrix} A_{k+1}(k,:) \\ A_{k+1}(n,:) \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} * \begin{bmatrix} A_k(k,:) \\ A_k(n,:) \end{bmatrix}$$

Siden $A_k(k, 1:k-1) = A_k(n, 1:k-1) = 0$ og elementet $A_{k+1}(n,k)$ nulles ut blir $A_{k+1}(n, 1:k) = 0$, mens fortsatt $A_{k+1}(k, 1:k-1) = 0$. Det følger at under diagonalen i A_{k+1} er det kun elementene $a_{n,k+1}, \ldots, a_{n,n}$ som kan være forskjellig fra null. Ved induksjon får vi at A_n er øvre triangulær.

(Fortsettes på side 4.)

Vi har $c = b_n$ hvor $b_1 = b$ og $b_{k+1} = R_k b_k$ for $k = 1, \ldots, n-1$. Vi kan beregne b_{k+1} fra b_k samtidig som vi beregner A_{k+1} fra A_k. Vi kan foreta oppdateringen av A_k og b_k samtidig ved

$$\begin{bmatrix} A_{k+1}(k, k:n) & b_{k+1}(k) \\ A_{k+1}(n, k:n) & b_{k+1}(n) \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} * \begin{bmatrix} A_k(k, k:n) & b_k(k) \\ A_k(n, k:n) & b_k(n) \end{bmatrix}$$

I det følgende Matlab programmet legger vi b i kolonne $n+1$ i A.

```
function x=rotsolve(A,b) n=length(b); A=[A b]; for k=1:n-1
    [c,s]=rot(A(k,k),A(n,k));
    A([k,n],k:n+1)=[c s; -s c]*A([k,n],k:n+1)
end x=utrisolve(A(:,1:n),A(:,n+1));
```

3c

Vi kunne også løst vårt spesielle system $Ax = b$ på andre måter. Vi skal her sammenligne med Gausseliminasjon uten pivotering. Vi trenger da bare å nulle ut elementene i rad n. Diskuter fordeler og ulemper ved metodene basert på rotasjoner og Gausseliminasjon.

Løsning Hovedfordelen med rotasjons-metoden er at den er ubetinget numerisk stabil. Den er numerisk stabil fordi vi foretar ortogonale transformasjoner og disse endrer ikke 2-normen til en vektor. Gausseliminasjon uten pivotering kan bryte sammen. Hvis vi må pivotere blir programmeringen mye mer komplisert og regnearbeidet øker. Både Gausseliminasjon og rotasjonsmetoden har kompleksitet $O(n^2)$, men Gausseliminasjon vil være raskere. Antall operasjoner for rotasjonsmetoden er $O(3n^2)$, mens Gausseliminasjon uten pivotering krever $O(n^2)$.

Oppgave 4 Sammenhengen mellom diskret Fourier transform og diskret sinustransform

Den diskrete Fourier transformen $d \in \mathbb{C}^N$ av en vektor $y \in \mathbb{C}^N$ er gitt ved

$$d = \frac{1}{N} F_N y, \quad \text{hvor} \quad F_N = \left(\omega_N^{(j-1)(k-1)}\right)_{k,j=1}^{N} \in \mathbb{C}^{N,N}$$

og

$$\omega_N = \exp^{-2\pi i/N} = \cos(\frac{2\pi}{N}) - i\sin(\frac{2\pi}{N}).$$

Her er $i = \sqrt{-1}$. Den diskrete sinustransformen $y \in \mathbb{R}^m$ av en vektor $x \in \mathbb{R}^m$ er gitt ved

$$y = S_m x, \quad \text{hvor} \quad S_m = \left(\sin\left(\frac{kj\pi}{m+1}\right)\right)_{k,j=1}^{m} \in \mathbb{R}^{m,m}.$$

(Fortsettes på side 5.)

La $x \in \mathbb{R}^m$ for en $m \geq 2$ og definer $z \in \mathbb{R}^{2m+2}$ ved

$$z = (0, x_1, \dots, x_m, 0, -x_m, -x_{m-1}, \dots, -x_1)^T \in \mathbb{R}^{2m+2}.$$

For eksempel er

$$F_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega_4 & \omega_4^2 & \omega_4^3 \\ 1 & \omega_4^2 & \omega_4^4 & \omega_4^6 \\ 1 & \omega_4^3 & \omega_4^6 & \omega_4^9 \end{bmatrix}$$

og

$$S_4 = \begin{bmatrix} \sin(\alpha) & \sin(2\alpha) & \sin(3\alpha) & \sin(4\alpha) \\ \sin(2\alpha) & \sin(4\alpha) & \sin(6\alpha) & \sin(8\alpha) \\ \sin(3\alpha) & \sin(6\alpha) & \sin(9\alpha) & \sin(12\alpha) \\ \sin(4\alpha) & \sin(8\alpha) & \sin(12\alpha) & \sin(16\alpha) \end{bmatrix}$$

hvor $\alpha = \pi/(m+1)$.

Du skal her vise følgende sammenheng mellom komponent k av $S_m x$ og komponent $k+1$ av $F_{2m+2}z$:

$$(S_m x)_k = \frac{i}{2} \left(F_{2m+2} z\right)_{k+1}, \quad k = 1, \dots, m.$$

Løsning *La* $\omega = \omega_{2m+2} = e^{-2\pi i/(2m+2)} = e^{-\pi i/(m+1)}$. *Rad* $k+1$ *av* $F_{2m+2}z$ *er gitt ved*

$$\begin{aligned} (F_{2m+2}z)_{k+1} &= \sum_{j=1}^{m} x_j \omega^{jk} - \sum_{j=m+2}^{2m+1} x_{2m+2-j}\omega^{jk} \\ &= \sum_{j=1}^{m} x_j \omega^{jk} - \sum_{j=1}^{m} x_j \omega^{(2m+2-j)k} \\ &= \sum_{j=1}^{m} x_j (\omega^{jk} - \omega^{-jk}) \\ &= -2i \sum_{j=1}^{m} x_j \sin\left(\frac{jk\pi}{m+1}\right) = -2i(S_m x)_k. \end{aligned}$$

Dividerer vi begge sider med $-2i$ *får vi resultatet.*

slutt