# MEK3570 Oblig 1

Soran Hussein Mohmmed / soranhm

September 2017

## Oppgave 1

Lar **A** være en tensor med matrisen [**A**] og $\mathbf{A}^T$ være den transponerte til **A** med matrisen $[\mathbf{A}]^T$. skal vise $[\mathbf{A}^T] = [\mathbf{A}]^T$.Det at T står innen for parantes betyr at man transponerer A før man setter opp som en matrise mens i den andre transponerer man etter at man har satt det opp som en matrise. Det jeg kom fram til ved hjelp av å tegne opp og et eksmepel var:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, A^T = \begin{pmatrix} a_{11} & a_{21} & a_{21} \\ a_{13} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{pmatrix}$$

$$[{a_{ij}}^T] = [a_{ij}]^T$$

$$a_{ji} = a_{ji}$$

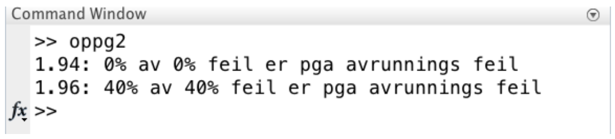
Eksempel:

$$[{a_{12}}^T] = [a_{12}]^T \rightarrow a_{12} = a_{12}$$

## Oppgave 2

Bekrefter ligningen 1.94 og 1.96 numerisk ved å test på matlab.
1.94: $\mathbf{I} : \mathbf{A} = tr(\mathbf{A}) = \mathbf{A} : \mathbf{I}$
1.96: $\mathbf{A} : (\mathbf{u} \otimes \mathbf{v}) = \mathbf{u} \cdot \mathbf{A}\mathbf{v} = (\mathbf{u} \otimes \mathbf{v}) : \mathbf{A}$
Bruker nå matlab med 100 random matriser og tester om ligningen 1.94 og 1.96 stemmer. Bruker en abs = $10^{-15}$ for å se hvor mange av feilene er pga avrundingsfeil. Kommer fram til at 1.94 og 1.96 stemmer men 1.96 foresaker små avrundingsfeil på 40-50%. Dette kommer av matlab runder ikke får med alle desimalene så den runder opp/ned. Jeg har brukt ferdige funksjoner i matlab, men dette kan også gjøres med for løkker der jeg setter hver verdi osv.

```
% Beviser 1.94 numerisk: (tester med 100 random matriser)
teller3 = 0;
teller4 = 0;
for s = 1:100
    abs = 10^-15;
    I = eye(3);
    A = rand(3,3);
    B = trace(A);
    C = sum(dot(I,A));
    D = sum(dot(A,I));

    if B ~= C || C ~= D || B ~= D
        if (B-C) < abs
            teller4 = teller4 +1;
        end
        teller3 = teller3 + 1;
    end
end
fprintf('1.94: %i%% av %i%% feil er pga avrunnings feil \n',teller4,teller3)

% Beviser 1.96 numerisk: (tester med 100 random matriser)
teller = 0;
teller2 = 0;
for s = 1:100
    abs = 10^-15;
    u = rand(1,3);
    v = rand(3,1);
    w = kron(u,v);
    A2 = rand(3,3);
    w2 = transpose(w);
    B = A2*v;

    C1 = sum(dot(A2,w2));
    C2 = dot(u,B);
    C3 = sum(dot(w2,A2));

    if C1 ~= C2 || C1 ~= C3 || C2 ~= C3
        if (C2-C1) < abs || (C1-C3) < abs || (C2-C3) < abs
            teller2 = teller2 +1;
        end
        teller = teller + 1;
    end

end
fprintf('1.96: %i%% av %i%% feil er pga avrunnings feil',teller2,teller)
```



```
>> oppg2
1.94: 0% av 0% feil er pga avrunnings feil
1.96: 40% av 40% feil er pga avrunnings feil
fx >>
```

Figure 1: resultatet etter kjøring av OPPGAVE 2

## Oppgave 3

Skal vertifisere:

$$grad(uv) = grad(u)v + grad(v)u, \quad div(u \times v) = v \cdot curlu - u \cdot curlv$$

Bruker innbygde funksjoner i matlab til å vise den første delen. Bruker u = sin(x) og v = cos(x) og holder meg unna randbetingalesene, altså bare tar hensyn til verdiene fra 21

```
start = 0;
slutt = 1;
n = 100;
h = (slutt−start)/n; % step
x = start:h:slutt;
y = start:h:slutt;
u = zeros(n,n);
v = zeros(n,n);

for i = 1:n
    for j = 1:n
        u(i,j) = sin(x(j));
        v(i,j) = cos(x(j));
   end
end
uv = u.*v;
[dudx,dudy] = gradient(u,h,h);
[dvdx,dvdy] = gradient(v,h,h);
[duvdx,duvdy] = gradient(uv,h,h);
dudx = dudx(21:80,21:80);
duvdx = duvdx(21:80,21:80);
dvdx = dvdx(21:80,21:80);
v = v(21:80,21:80);
u = u(21:80,21:80);
dudy = dudy(21:80,21:80);
duvdy = duvdy(21:80,21:80);
dvdy = dvdy(21:80,21:80);


% grad(u*v) = grad(u) * v + u * grad(v)

lhsi = sqrt((duvdx − v.*dudx − u.*dvdx).^2);
lhsj = sqrt((duvdy − v.*dudy − u.*dvdy).^2);
abs = 10^−4; teller = 0; abs2 = 10^−6; teller2 = 0;
for i = 1:60
    for j = 1:60
        if lhsi(i,j) < abs && lhsj(i,j) < abs
            teller = teller + 1;
        end
        if lhsi(i,j) < abs2 && lhsj(i,j) < abs2
            teller2 = teller2 + 1;
        end
    end
end
riktig = (teller/(60*60)) * 100;
riktig2 = (teller2/(60*60)) * 100;

fprintf('Dette gir %.2f %% riktig, med en abs = %.1e og \nDette gir %.2f%% riktig, med en abs

%  || grad(u*v) - grad(u) * v - u * grad(v)||
```

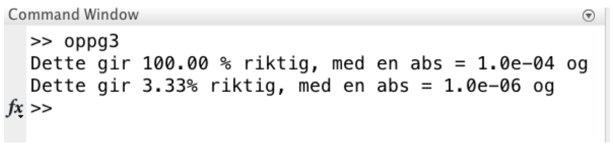
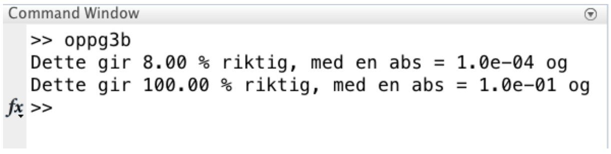

```
Command Window
>> oppg3
Dette gir 100.00 % riktig, med en abs = 1.0e-04 og
Dette gir 3.33% riktig, med en abs = 1.0e-06 og
fx >>
```

Figure 2: resultatet etter kjøring av OPPGAVE 3a

For å teste divergens og virvling velger jeg en tilfeldig 3x3 matrises, som igjen gir meg en 4x4 matrise til slutt pga matrise utregning.

```
%div = divergence(dot(u,v),x)
%ku = v * curl(sin(u)) - u * curl(v)
n = 100;
x = linspace(0,1,n);
y = linspace(0,1,n);
z = linspace(0,1,n);
[X,Y,Z] = meshgrid(x,y,z);

dx = x(2) - x(1);
dy = y(2) - y(1);
dz = z(2) - z(1);

u = zeros(n,n);
v = zeros(n,n);


for i = 1:length(x)
    for j = 1:length(y)
        for k = 1:length(z)
            u(i,j,k,1) = sin(x(i));
            u(i,j,k,2) = sin(2*x(k));
            u(i,j,k,3) = cos(2*x(j));
            v(i,j,k,1) = sin(3*x(k));
            v(i,j,k,2) = cos(x(i));
            v(i,j,k,3) = cos(x(j));
        end
    end
end

u_K_v = cross(u,v);
[curl_u(:,:,:,1),curl_u(:,:,:,2),curl_u(:,:,:,3),cav] = curl(X,Y,Z,u(:,:,:,1),u(:,:,:,2),u(:,:
[curl_v(:,:,:,1),curl_v(:,:,:,2),curl_v(:,:,:,3),cav] = curl(X,Y,Z,v(:,:,:,1),v(:,:,:,2),v(:,:
```

```
lhs = divergence(X,Y,Z,u_K_v(:,:,:,1),u_K_v(:,:,:,2),u_K_v(:,:,:,3));

rhs1 = dot(v,curl_u,4);
rhs2 = dot(u,curl_v,4);
rhs = rhs1-rhs2;

test = sqrt((lhs-rhs).^2);
abs = 10^-4; abs2 = 10^-1;
teller = 0; teller2 = 0;
for i = 1:n
    for j = 1:n
        for k = 1:n
            if test(i,j,k) < abs
                teller = teller +1 ;
            end
            if test(i,j,k) < abs2
                teller2 = teller2 +1 ;
            end
        end
    end
end
 riktig = teller/1000000 * 100;
 riktig2 = teller2/1000000 * 100;
 fprintf('Dette gir %.2f %% riktig, med en abs = %.1e og \nDette gir %.2f %% riktig, med en ab
```


Command Window

```
>> oppg3b
Dette gir 8.00 % riktig, med en abs = 1.0e-04 og
Dette gir 100.00 % riktig, med en abs = 1.0e-01 og
>>
```

Figure 3: resultatet etter kjøring av OPPGAVE 3b

## Oppgave 4

I en bestemt område er den romlige hastighetsfeltet$\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$ er gitt som:

$$v_1 = -\alpha(x_1^3 + x_1 x_2^2)e^{-\beta t}, \quad v_2 = \alpha(x_1^2 x_2 + x_2^3)e^{-\beta t}, \quad v_3 = 0$$

hvor $\alpha, \beta > 0$ er gitt konstant. Finn komponenten i den romslige akkselrasjonsfeltet $\mathbf{a} = \mathbf{a}(\mathbf{x}, t)$ i punktet (1,0,0) og tiden t = 0.

$$a_1 = \frac{dv_1}{dt} = \alpha\beta(x_1^3 + x_1 x_2^2)e^{-\beta t}, \quad a_2 = \frac{dv_2}{dt} = -\alpha\beta(x_1^2 x_2 + x_2^3)e^{-\beta t}, \quad a_3 = 0$$

i punktet (1,0,0) og tiden t = 0:

$$a_1 = \alpha\beta(1^3 + 1 * 0^2)e^{-\beta * 0} = \alpha\beta, \quad a_2 = -\alpha\beta(1^2 * 0 + 0^3)e^{-\beta * 0} = 0, \quad a_3 = 0$$