
Innlevering B (sindre pedersen thuv, Johannes Foslund, Fredrik Eilertsen, Morten Bertheussen)

Table of Contents

Oppgave 1	1
Oppgave 1 a)	2
Oppgave 1 b)	3
Oppgave 1 c)	4
Oppgave 1 d)	5
Oppgave 2	6
Oppgave 3	7
Oppgave 4	8
A)	8
B)	9
C)	10

Oppgave 1

`%Trekanten ?ABC er gitt ved A(2, 0), B(?1, 3) og C(?2, ?4).`

`%Tegne trekanten`

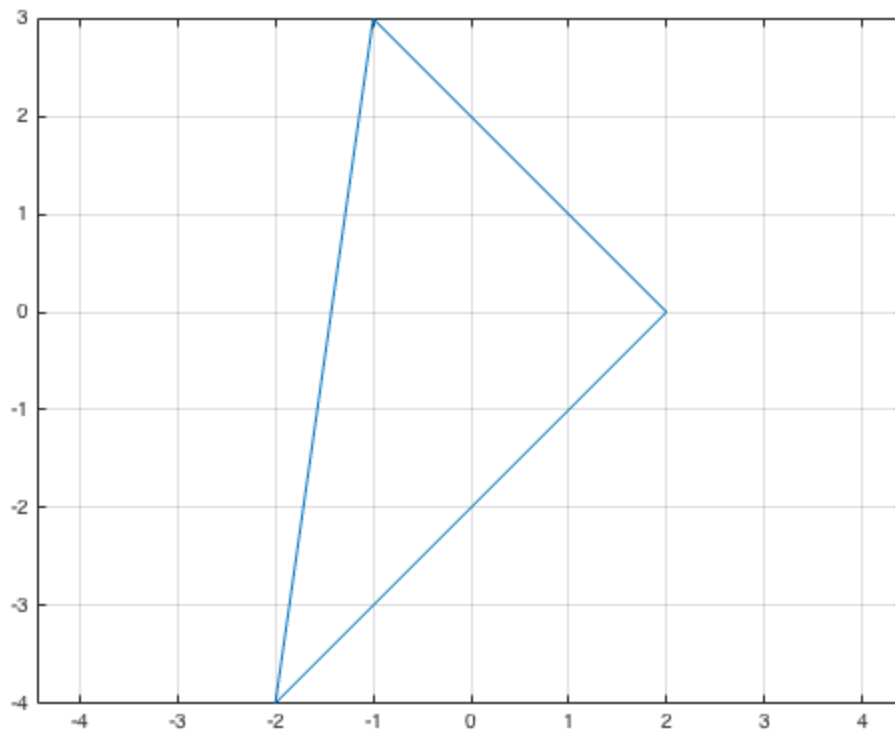
`A = [2 ; 0];`

`B = [-1 ; 3];`

`C = [-2 ; -4];`

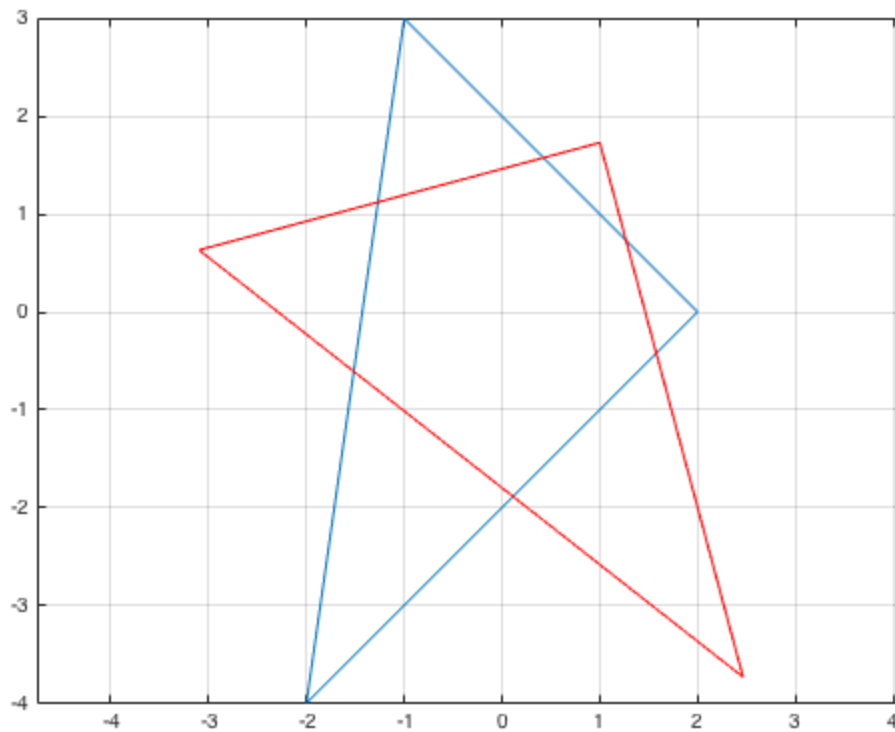
`ABC = [A B C A]; % trekant ABC`

`plot(ABC(1,:), ABC(2,:)), axis equal, grid on, hold on`



Oppgave 1 a)

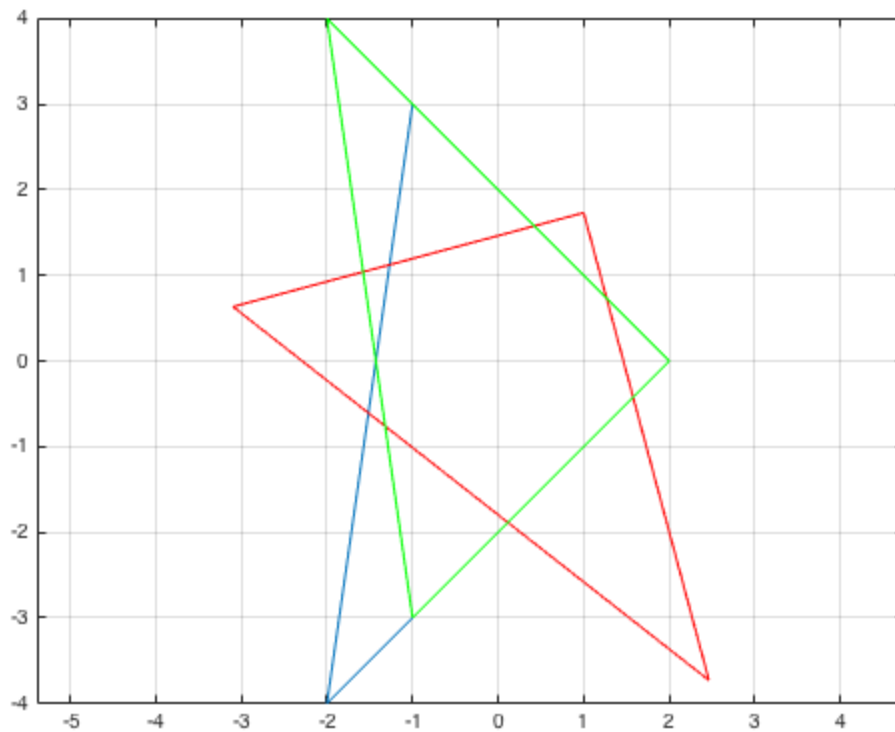
```
%) Roter trekanten med 60 grader  
% i positiv omløpsretning om origo (mot klokka),  
% og tegn den roterte trekanten i samme koordinatsystem som den  
% originale  
% trekanten.  
  
% Rotasjonsmatrise  
theta = 60;  
Mrot = [ cosd(theta) -sind(theta)  
        sind(theta) cosd(theta) ];  
  
ABC_rotert = Mrot*ABC;  
plot(ABC_rotert(1,:), ABC_rotert(2,:), 'r')
```



Oppgave 1 b)

Spegle den originale trekanten om y-aksen. Tegn den speglete trekanten i samme koordinatsystem som trekantene i oppgave a.

```
% Speilingsmatrise  
Mspeil = [1 0  
          0 -1];  
ABC_speil = Mspeil*ABC;  
plot(ABC_speil(1,:), ABC_speil(2,:), 'g')
```

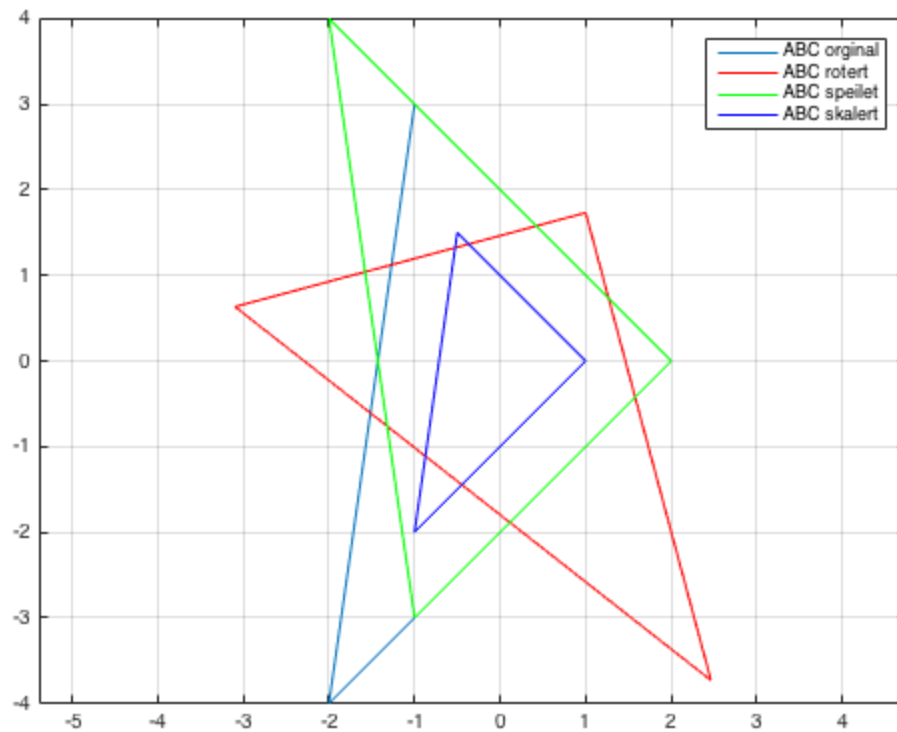


Oppgave 1 c)

Skaler den originale trekanten til halv størrelse, og tegn den skalerte trekanten i samme figur som figuren som i oppgave a. Legg inn legend som viser hvilke trekanter det er snakk om

```
% Skaleringsmatrise
k = 0.5;
Mskalar = [ k 0
            0 k ];

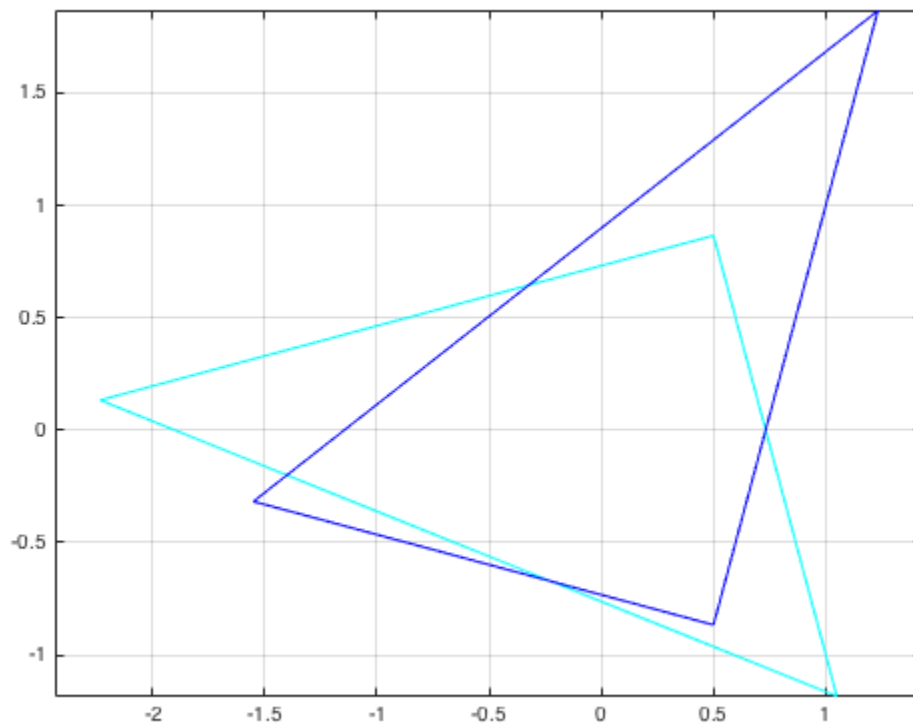
ABC_skalert = Mskalar*ABC;
plot(ABC_skalert(1,:), ABC_skalert(2,:), 'b')
legend('ABC orginal', 'ABC rotert', 'ABC speilet', 'ABC skalert')
```



Oppgave 1 d)

```
ABC_1 = Mrot*Mspeil*Mskalar*ABC;  
  
ABC_2 = Mrot*Mskalar*Mspeil*ABC;  
  
ABC_3 = Mskalar*Mrot*Mspeil*ABC;  
  
ABC_4 = Mskalar*Mspeil*Mrot*ABC;  
  
ABC_5 = Mspeil*Mskalar*Mrot*ABC;  
  
ABC_6 = Mspeil*Mrot*Mskalar*ABC;  
  
% Vi definerer hjelpefunksjonen plotObject  
plotObject = @(Object,Spec) plot(Object(1,:),Object(2,:),Spec);  
figure(2)  
plotObject(ABC_1,'y'), axis equal, grid on, hold on  
plotObject(ABC_2,'m')  
plotObject(ABC_3,'c')  
plotObject(ABC_4,'r')  
plotObject(ABC_5,'g')  
plotObject(ABC_6,'b'), hold off  
  
%Vi får 2 ulike trekanter. Dette er når Mrot og Mspeil bytter plass.
```

%Mskalar er lik for alle.



Oppgave 2

eulers_metode.m

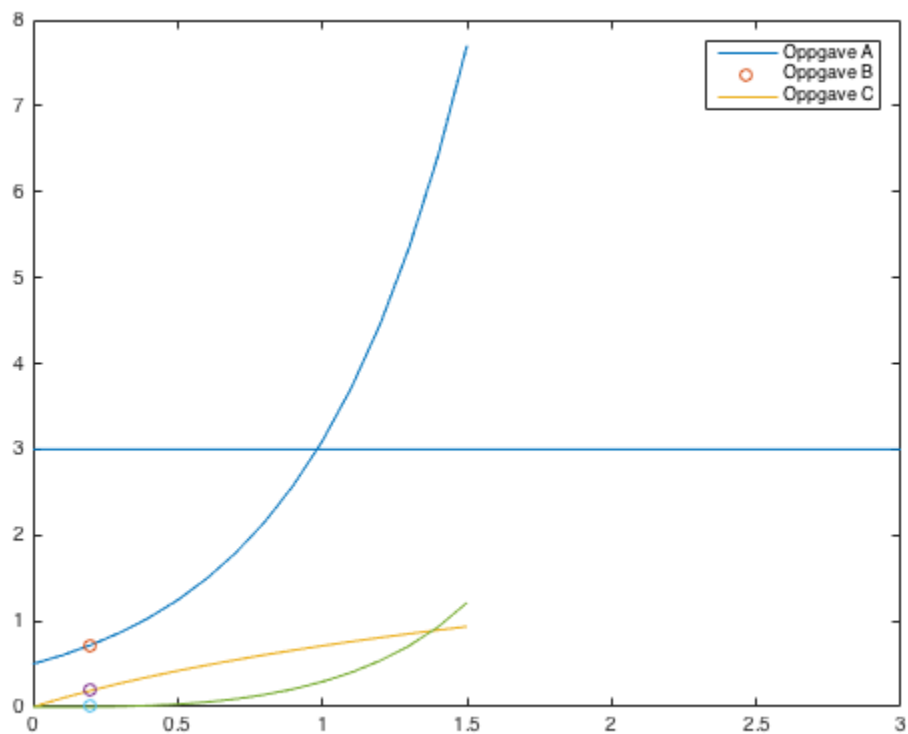
```
clear
clear x y
dx = 0.1; % steglengde
F = @(x,y) 2*y; % høgresiden i differensiallikningen
G = @(x,u) 1/exp(u); % høgresiden i differensiallikningen
H = @(x,z) x^2 + z^2; % høgresiden i differensiallikningen
x(1) = 0; % initialbetingelsen y(0) = 1
y(1) = 0.5; % => x = 0 og y = 1
u(1) = 0; % => x = 0 og y = 1
z(1) = 0; % => x = 0 og y = 1
i=3;
for n = 1:15
    x(n+1) = x(n) + dx;
    y(n+1) = y(n) + F(x(n),y(n))*dx;
    u(n+1) = u(n) + G(x(n),u(n))*dx;
    z(n+1) = z(n) + H(x(n),z(n))*dx;
end

plot(x,y,x(3),y(3),'o'), hold on
```

```
plot(x,u,x(3), u(3), 'o')  
plot(x,z,x(3), z(3), 'o')  
line([0 3],[3 3]), hold off  
legend('Oppgave A','Oppgave B','Oppgave C')
```

```
display('Oppgave 1')  
fprintf('Oppgave A er %0.2f \n', y(3))  
fprintf('Oppgave B er %0.2f \n', u(3))  
fprintf('Oppgave C er %0.3f \n', z(3))  
display('-----')  
display('Oppgave 2 vises grafisk')
```

```
Oppgave 1  
Oppgave A er 0.72  
Oppgave B er 0.19  
Oppgave C er 0.001  
-----  
Oppgave 2 vises grafisk
```



Oppgave 3

```
%function [ Kvadratrotten ] = innlevering_B( number ) %starten av  
%funksjonen  
  
number = 75; % eksempel nummer inn i funksjonen. Ellers ville du  
skrevet innlevering_B(75)
```

```
if number > 0                                %hvis funksjonen får in et positivt tall kjør

    start = double(0);
    slutt = double(number);
    midt = double((start+slutt)/2);
    temp_mid = double(start);

while number > midt

    if (midt*midt) > number

        slutt = midt;
    else

        start = midt;
    end

    temp_mid = midt;
    midt = (start+slutt)/2;
    differanse = number - (midt*midt);

    if differanse >= 0 && differanse < 0.0005

        break;
    end

end

format long;
Kvadratrotten = midt;

disp(Kvadratrotten);

else

    fprintf('Error: her må du bruke positive tall\n');

end

%end slutt på funksjonen

8.660244941711426
```

Oppgave 4

A)

```
close all;
```



```
%7.13

snittflate = [ 1 2 3 4 5 6 7 8 9;
              0.3 22.7 48.8 73.2 88.4 82.8 58.7 26.2 3.9 ];

intervall = size(snittflate);
intervall = intervall(1,2);

n = (intervall-1)/2;

a = snittflate(1,1);
b = intervall(end);

simsum = 0;

for i = 2:(intervall-1)

    if mod(snittflate(1,i),2) == 0
        k = 4;
    else
        k = 2;
    end

    simsum = simsum + (k * snittflate(2,i));
end

Areal = (((b-a)/(6 * n)) * (snittflate(2,1) + simsum +
    snittflate(2,end)));

Areal = Areal * 10;

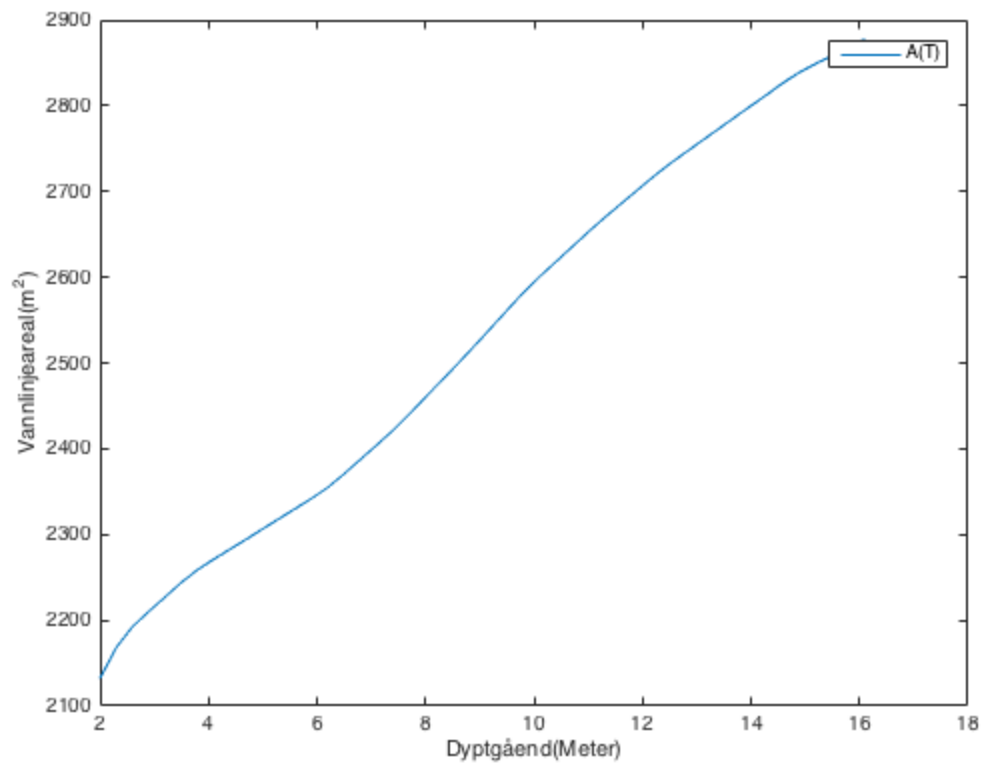
fprintf('7.13) \n Arealet av båten = %0.2f m^3 \n', Areal);

7.13)
    Arealet av båten = 4052.00 m^3
```

B)

```
close all;
load 'tankskip.mat';

plot(T(:,1),A(:,1));
hold on;
legend('A(T)',2);
hold on;
xlabel('Dyptgåend(Meter)');
hold on;
ylabel('Vannlinjeareal(m^2)');
hold off;
```



C)

```
clear all;
load('tankskip.mat');

A_trapes = 0;
a = 2;      %start
b = 11;     %slutt

k = size(T,1); %finner størrelsen på matrisen og lagrer den i 1
dimensjon

for i = 1:k      % finner plasseringen til sluttverdi
    if T(i,1) == b
        b = i;
        break;
    end
    if T(i,1) == a % finner plasseringen til startverdi
        a = i;
    end
end

h = T(2)-T(1); %finner høyden
```

```
for i = a:b % fra start til slutt
    A_trapes = A_trapes + h/2*(A(i) + A(i+1));
end

A_trapes = A_trapes + V2(1,1);

format long;
disp(A_trapes);

2.616741750960258e+04
```

Published with MATLAB® R2015b

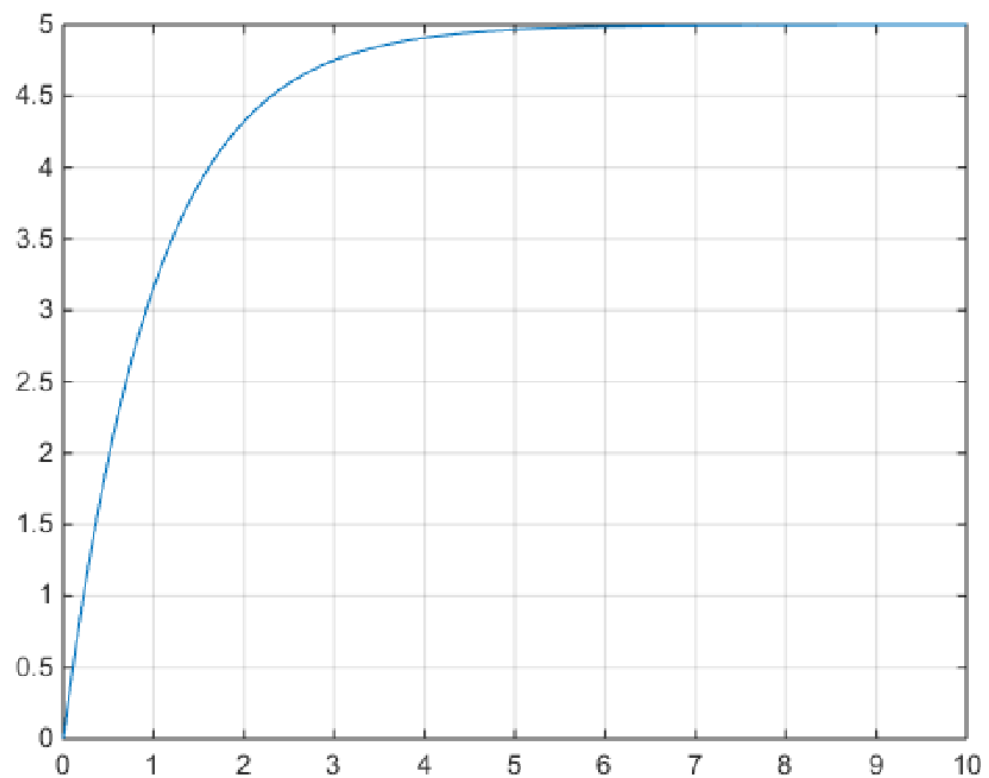
Oppgave EL-LÆRE a)

```
clear all
R = 10000;           % ohm
C = 100*10^-6;       % F
U = 5;               % V
Tau = R*C

tid = 0:.01:10;
Uc = U*(1 - exp(-tid/(Tau))); % Spenningen over kondensatoren
plot(tid,Uc), grid on , hold on
```

$\tau =$

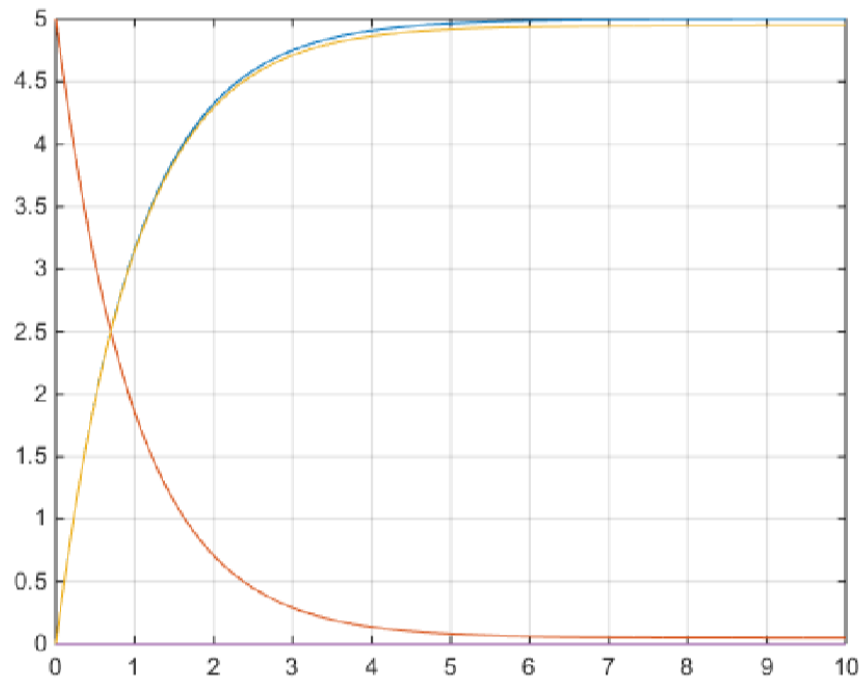
1.0000



Oppgave EL-LÆRE b)

```
Ir = csvread('oppladning.csv',0,2); % Henter inn strømmen over resista
Spenningc = csvread('oppladning.csv',0,1); % Henter inn spenningen over konde
tiden = csvread('oppladning.csv',0,0); % Henter inn tiden fra fila

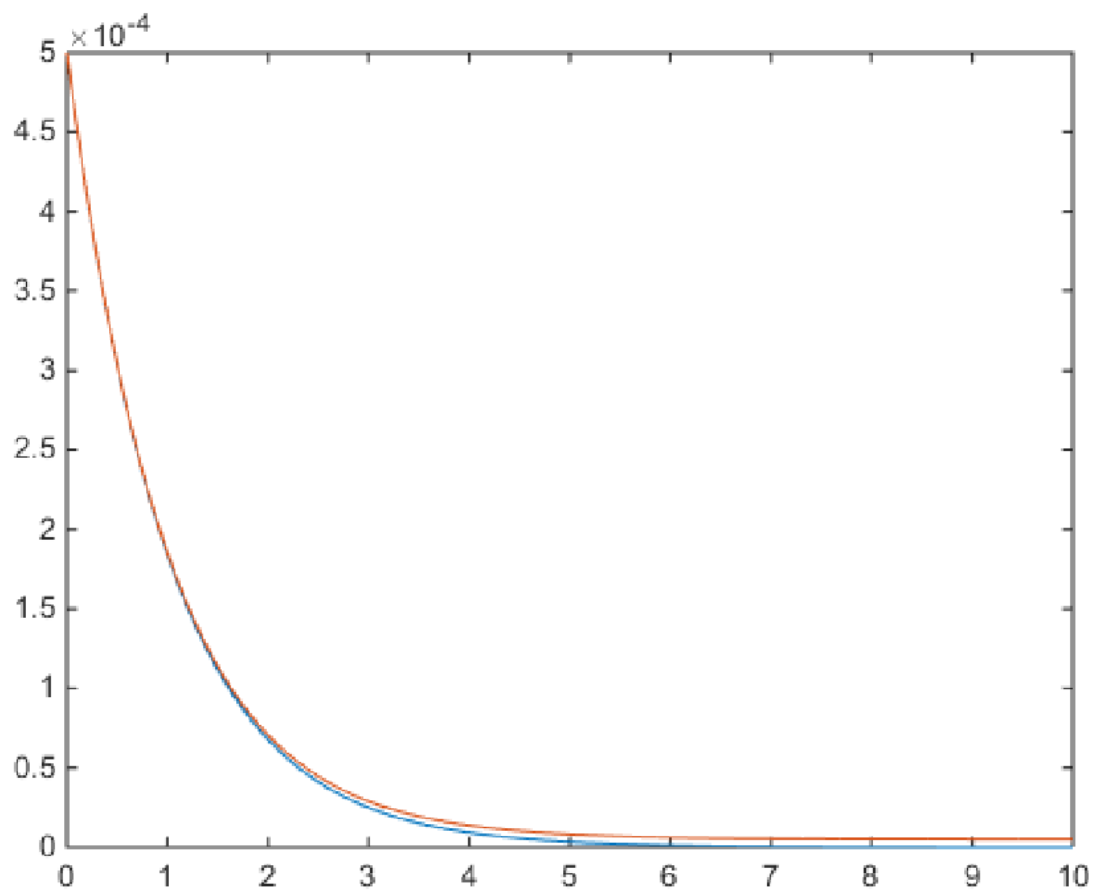
x=0:0.01:10;
plot(x,Ir*10000)
plot(x,Spenningc), hold off
```



Oppgave EL-LÆRE c)

```
clear x

U = 5;                                % V
Irt = (U-Uc)/R;                       % Irt = teoretisk strøm over resistansen
x=0:0.01:10;
plot(x,Irt), hold on
plot(x,Ir), hold off
```



Oppgave EL-LÆRE d)

```
clear b n a areal Ir

a = 0;
b = 10;
n = 5;

Ir=@(x) (U/R)* exp(-x/l);
areal = ((b-a)/(6*n))*(Ir(0)+4*Ir(1)+2*Ir(2)+4*Ir(3)+2*Ir(4)+4*Ir(5)+2*Ir(6)+4*Ir(7)+Ir(8));

ladning = (5/10000)*1    %C

avvik = (1-(ladning/areal))*100;
fprintf('Avviket mellom teoretisk og målt ladning er %d prosent',avvik)

areal =

    5.0246e-04

ladning =

    5.0000e-04

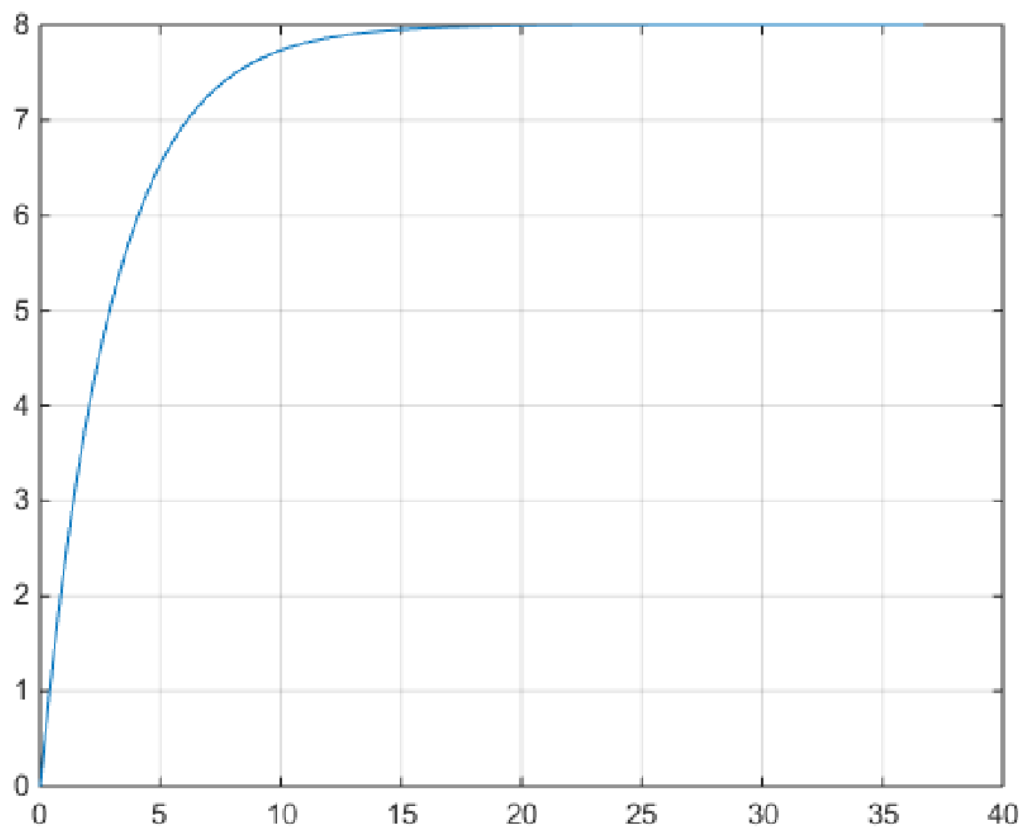
Avviket mellom teoretisk og målt ladning er 4.887550e-01 prosent
```

Oppgave EL-LÆRE e)

```
clear all
R = 4660;           % ohm
C = 631*10^-6;      % F
U = 8;              % V
Tau = R*C;

tid = 0:0.1:36.7;
Uc = U*(1 - exp(-tid/(Tau)));

plot(tid,Uc),grid on, hold on
```

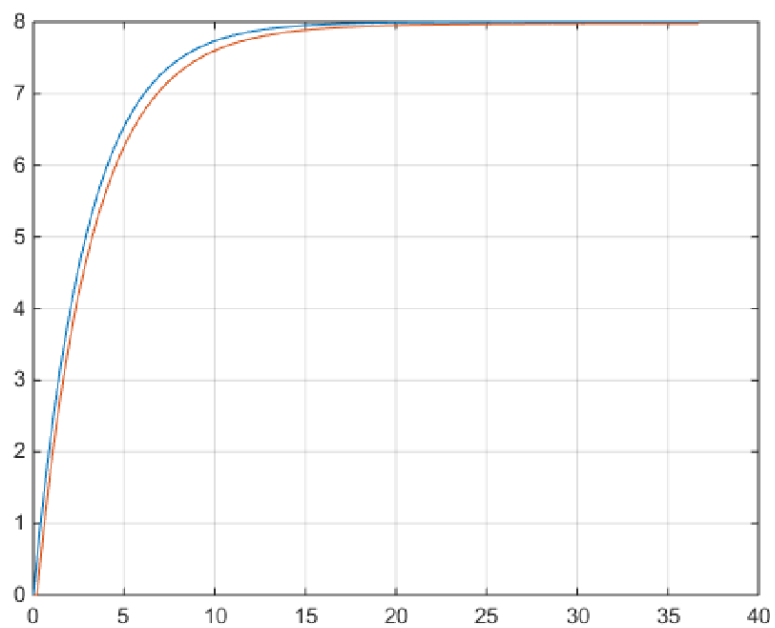


Oppgave EL-LÆRE e)

```
R = 4700 %ohm
U = 8 %V
C = 680*10^(-6) %F

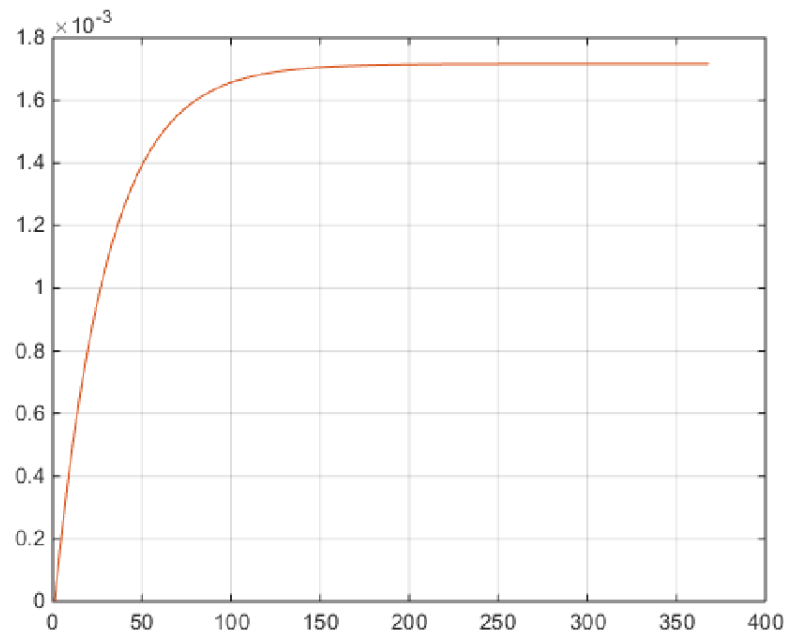
Ur = xlsread('raadata.xlsx','E:E'); % Importerer data fra våres egen fil

x=0:0.1:36.7;
plot(x,Ur), hold off
```



Oppgave EL-LÆRE e)

```
Im = Uc/4660; % Im er målt strøm  
plot (Im), grid on, hold on  
plot(Uc/4660), hold off
```



Oppgave EL-LÆRE e)

```
clear areal x i avvik
simp1 = 10/(6*(367/2));
simp = 0;

simp2 = 0;
for x=2:2:366;
    simp = simp + 4*(Uc(x)/4660);      % Regner partallene
end
for i=3:2:365;
    simp2 = simp2 + 2*(Uc(i)/4660);    % Regner oddetallene
end
areal = simp1*((Uc(1)/4660)+simp+simp2+(Uc(367)/4660)) % Legger sammen

teoretiskQ = (8/4660)*10;
avvik = (1-(teoretiskQ/areal))*100;
fprintf('Avviket mellom teoretisk og målt ladning er %d prosent',avvik)

areal =

    0.0157

Avviket mellom teoretisk og målt ladning er -9.032944e+00 prosent
```

Oppgave EL-LÆRE f)

Med

$A = 5;$

$F = 3;$

$U_0 = 1;$

