Løsningsforslag eksamensopgaver M3NUM1 S20

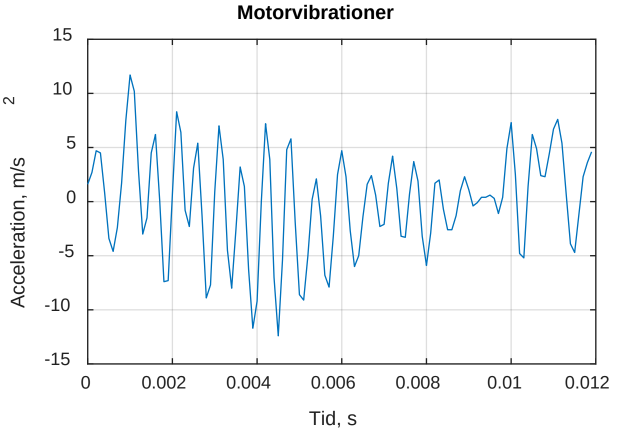
OPGAVE 1

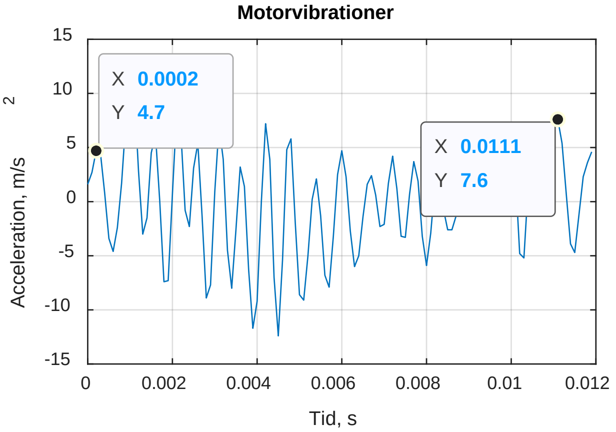
(1a)

Her følger MATLAB-koden til frembringelse af plottet:

```
a = [ 1.6   2.7   4.7   4.5   0.8  -3.4  -4.6  -2.4   1.7   7.5  11.7  10.2 ...
      2.8  -3.0  -1.5   4.5   6.2   0.1  -7.4  -7.3   0.7   8.3   6.4  -0.8 ...
     -2.3   3.1   5.4  -1.3  -8.9  -7.7   0.9   7.0   3.9  -4.5  -8.0  -2.6 ...
      3.2   1.4  -6.3 -11.7  -9.2   0.0   7.2   3.9  -7.0 -12.4  -5.4   4.8 ...
      5.8  -1.9  -8.6  -9.1  -5.0   0.2   2.1  -1.3  -6.8  -7.9  -3.2   2.5 ...
      4.7   2.3  -2.7  -6.0  -5.0  -1.4   1.6   2.4   0.6  -2.3  -2.1   1.7 ...
      4.2   1.2  -3.2  -3.3   0.7   3.7   1.9  -3.2  -5.9  -2.9   1.7   2.0 ...
     -0.7  -2.6  -2.6  -1.3   1.0   2.3   1.1  -0.4  -0.1   0.4   0.4   0.6 ...
      0.3  -1.1   0.4   4.9   7.3   2.5  -4.8  -5.2   1.4   6.2   4.9   2.4 ...
      2.3   4.4   6.7   7.6   5.4   0.7  -3.9  -4.7  -1.2   2.3   3.6   4.6];
n = length(a);
fs = 10000;
dt = 1/fs;
t = (0:n-1)*dt;
plot(t,a)
grid
title('Motorvibrationer')
xlabel('Tid (s)')
ylabel('Acceleration, m/s^2')
```

Herunder er det originale plot (t.v.) samt det samme plot (t.h.) påført markering af to toppe.

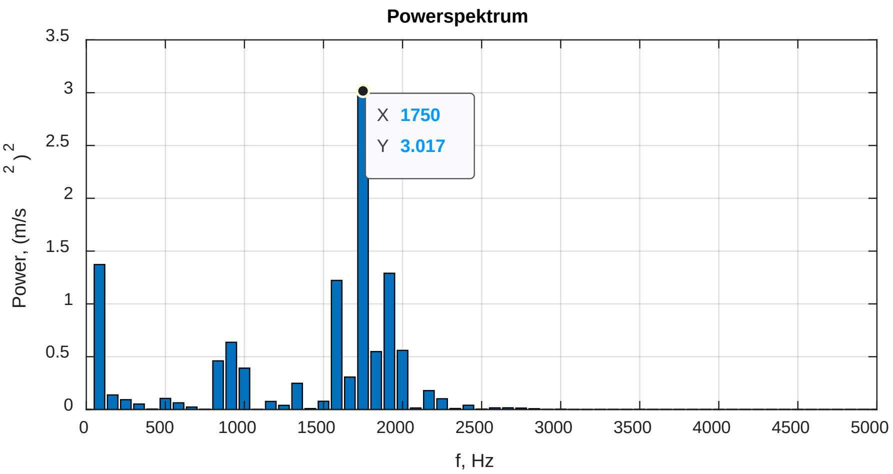




<u>Der synes at være en fremherskende frekvens</u>, idet der forekommer svingninger, der topper med regelmæssig tidsafstand. De to markerede toppe forekommer efter henholdsvis 0,0002 s og 0,0111 s, og ved optælling finder man, at der er 19 svingninger mellem toppene. Dette svarer til en fremherskende frekvens på 19/(0,0111-0.0002) = <u>1743 Hz</u>.

(1b)

Her følger et plot af tidsseriens powerspektrum samt MATLAB-koden, der har været benyttet til at beregne og plotte spektret (forudsætter, at MATLAB-koden under opg. 1a allerede er kørt):




```
T = n*dt;                % Tidsseriens længde (s)
Y = fft(a)/n;            % Fast Fouriertransformation
fmax = fs/2;             % Højeste detekterbare frekvens (Hz)
df = 1/T;                % Afstand mellem detekterbare frekvenser (Hz)
P = real(Y).^2 + imag(Y).^2;  % Beregn vektor af power-værdier
f = (0:n-1)*df;          % Frekvenser
figure
bar(f,P);
xlim([0 fmax]);          % Afgræns frekvensaksen
xlabel('f, Hz'); ylabel('Power, (m/s^2)^2');
title('Powerspektrum');
grid
```

Af powerspektret fremgår, at den højeste powerværdi opnås ved frekvensen 1750 Hz, som således er den dominerende frekvens.

(1c)

Det færdiggjorte script:

```
clear
f = 0:200:5000;
P = [0.00 0.26 0.24 0.65 1.97 0.32 0.04 0.00 0.90 ...
     1.44 3.60 0.08 0.58 0.06 0.09 0.10 0.04 0.06 ...
     0.06 0.05 0.05 0.05 0.05 0.05 0.04 0.05];
N = length(P);
fprintf('f, Hz   P, (m/s^2)^2\n')
for i = 2:N-1
  if P(i) > P(i-1) && P(i) > P(i+1)
    fprintf('%5.0f   %6.4f\n',f(i),P(i))
  end
end
```

Output i kommandovinduet:

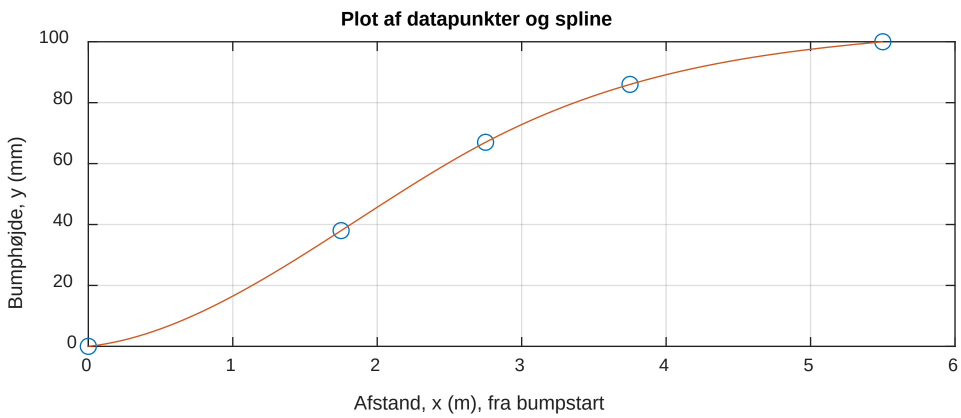
```
f, Hz   P, (m/s^2)^2
  200   0.2600
  800   1.9700
 2000   3.6000
```

```
2400    0.5800
3000    0.1000
```

OPGAVE 2

(2a)
Følgende MATLAB-kommandoer producerer det ønskede plot og *spline*-resultat, som vises efterfølgende.

```
x = [0 1.75 2.75 3.75 5.5]; % X-værdier ...
y = [0 38 67 86 100];       % ... og tilhørende y-værdier
xx = linspace(0,5.5);       % 100 x-værdier i intervallet fra 0 til 5,5 m
yy = spline(x,y,xx);        % De tilsvarende 100 y-værdier iflg. en kubiske spline
plot(x,y,'o',xx,yy)         % Plot af datapunkter og spline
title('Plot af datapunkter og spline')
xlabel('Afstand, x (m), fra bumpstart')
ylabel('Bumphøjde, y (mm)')
grid
y_4_75 = spline(x,y,4.75)   % Estimér y(4,75 m) ud fra spline
```

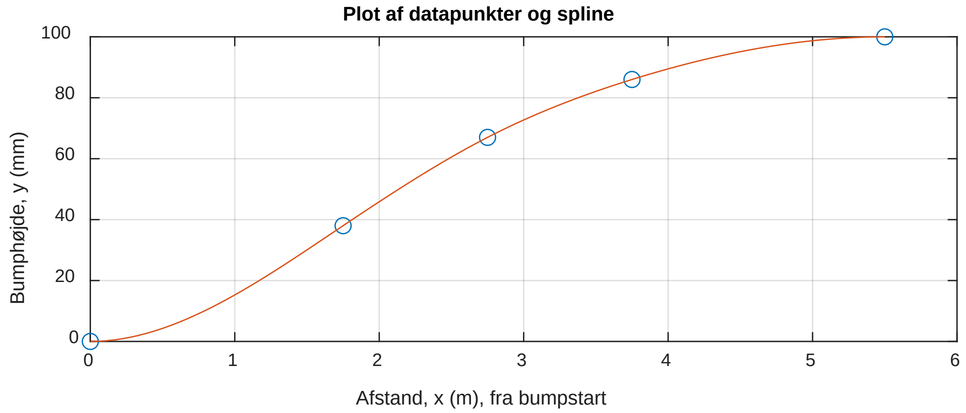



```
y_4_75 = 95.9710
```

Som det fremgår af beregningsresultatet, gælder der iflg. den kubiske *spline*, at $\underline{\underline{y(4{,}75\text{ m}) = 96\text{ mm}}}$.

(2b)
Følgende MATLAB-kommandoer producerer det ønskede plot og *spline*-resultat, som vises efterfølgende.

```
x = [0 1.75 2.75 3.75 5.5]; % X-værdier ...
y = [0 38 67 86 100];       % ... og tilhørende y-værdier
y_m_diff = [0 y 0];         % Udvidet y-vektor med 0 i enderne = vandret tangent
xx = linspace(0,5.5);       % 100 x-værdier i intervallet fra 0 til 5,5 m
yy = spline(x,y_m_diff,xx); % De tilsvarende 100 y-værdier iflg. en kubiske spline
plot(x,y,'o',xx,yy)         % Plot af datapunkter og spline
title('Plot af datapunkter og spline')
xlabel('Afstand, x (m), fra bumpstart')
ylabel('Bumphøjde, y (mm)')
grid
y_4_75 = spline(x,y_m_diff,4.75) % Estimér y(4,75 m) ud fra spline
```




```
y_4_75 = 97.1982
```

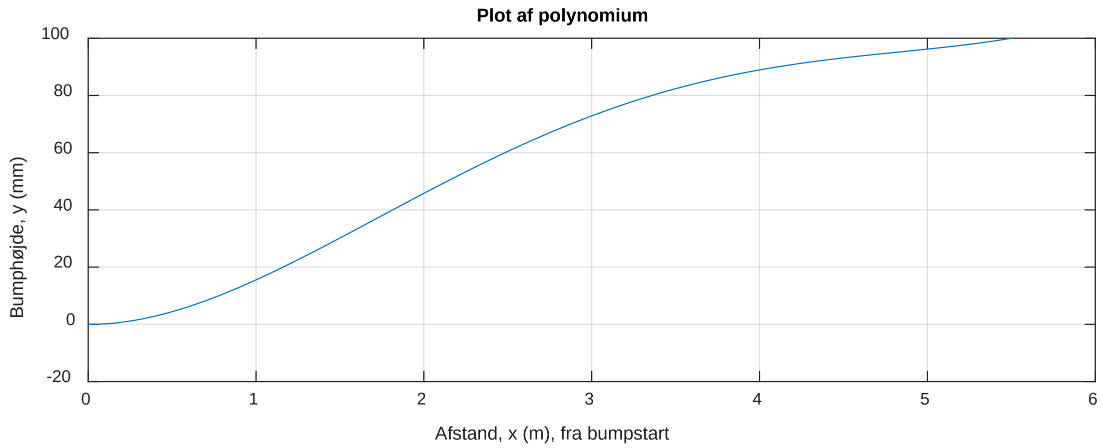
Som det fremgår af beregningsresultatet, gælder der iflg. den kubiske *spline*, at $\underline{\underline{y(4{,}75\text{ m}) = 97\text{ mm}}}$.

(2c)

Følgende MATLAB-kommandoer giver som output de beregnede polynomiumskoefficienter, det ønskede plot og den beregnede y-værdi ved $x = 4{,}75$ m.

```
x = [0 1.75 2.75 3.75 5.5]; % X-værdier ...
y = [0 38 67 86 100];       % ... og tilhørende y-værdier
p = polyfit(x,y,4)          % Bestem polynomiumskoefficinterne
xx = linspace(0,5.5);       % 100 x-værdier i intervallet fra 0 til 5,5 m
yy = polyval(p,xx);         % De tilsvarende 100 y-værdier iflg. polynomiet
plot(x,y,'o',xx,yy)         % Plot af datapunkter og polynomium
title('Plot af datapunkter og polynomium')
xlabel('Afstand, x (m), fra bumpstart')
ylabel('Bumphøjde, y (mm)')
grid
y_4_75 = polyval(p,4.75)    % Estimér y(4,75 m) ud fra polynomiet
```

```
p =
     0.4194   -5.4996   20.9235   -0.3069   -0.0000
```




```
y_4_75 = 94.7129
```

Som det fremgår af beregningsresultaterne, fås følgende polynomiumskoefficienter:

$$p_1 = 0{,}4194, \qquad p_2 = -5{,}4996, \qquad p_3 = 20{,}9235, \qquad p_4 = -0{,}3069, \qquad p_5 = 0{,}0000$$

Desuden estimeres bumphøjden ved $x = 4{,}75$ m til værdien 95 mm.

OPGAVE 3

(3a)
Vi samler alle led på venstresiderne i begge ligninger og definerer derefter funktioner, $f_1(x,y)$ og $f_2(x,y)$, der indeholder udtrykkene på ligningernes venstresider:

$$xa + (x-y)(by+c-y) - 2d = 0$$
$$s + xt^2 - (xu+y)v = w \Leftrightarrow s + xt^2 - (xu+y)v - w = 0$$

dvs.

$$f_1(x,y) = xa + (x-y)(by+c-y) - 2d$$
$$f_2(x,y) = s + xt^2 - (xu+y)v - w$$

Metode 1 til afgørelse af om ligningssystemet er lineært eller ulineært

Ligningssystemet er lineært, hvis og kun hvis begge funktioner kan skrives på formen

$$\text{konstant} \cdot x + \text{konstant} \cdot y + \text{konstant}$$

Funktionsudtrykkene ganges ud, og led med henholdsvis x og y samles:

$$f_1(x,y) = (b-1)\,x\,y + (a+c)\,x + (1-b)\,y^2 + (-c)\,y - 2\,d$$
$$f_2(x,y) = (t^2 - u\,v)\,x + (-v)\,y + s - w$$

Funktionen f_1 er ulineær, da der udtrykket indgår led indeholdende produkter af de ubekendte. Ligningssystemet er således ulineært.

Metode 2 til afgørelse af om ligningssystemet er lineært eller ulineært

Jacobimatricen, J, svarende til de to funktioner bestemmes ved at differentiere funktionerne partielt mht. x og y:

$$J = \begin{pmatrix} \frac{\partial}{\partial x} f_1(x,y) & \frac{\partial}{\partial y} f_1(x,y) \\ \frac{\partial}{\partial x} f_2(x,y) & \frac{\partial}{\partial y} f_2(x,y) \end{pmatrix} = \begin{pmatrix} a+c-y+b\,y & y-c-b\,y+(x-y)\,(b-1) \\ t^2 - u\,v & -v \end{pmatrix}$$

Da begge udtryk i øverste række i Jacobimatricen indeholder ubekendte (x og/eller y) er ligningssystemet ulineært.

(3b)
Indsættes $a = 2,\ b = 1,\ c = 5,\ d = 19,\ s = 6,\ t = 2,\ u = -4,\ v = -2$ og $w = -14$ i ligningssystemet fås:

$$7x - 5y = 38$$
$$-4x + 2y = -20$$

Da begge ligninger er lineære, dvs. af formen $\text{konstant} \cdot x + \text{konstant} \cdot y = \text{konstant}$, kan ligningssystemet opskrives på formen **AX = B**, hvor

$$\mathbf{A} = \begin{pmatrix} 7 & -5 \\ -4 & 2 \end{pmatrix},\ \mathbf{B} = \begin{pmatrix} 38 \\ -20 \end{pmatrix} \text{ og } \mathbf{X} = \begin{pmatrix} x \\ y \end{pmatrix}$$

Ligningssystemet løses vha. MATLAB:

```
A = [7 -5; -4 2]; B = [38; -20];
X = A\B;
x = X(1), y = X(2)
```

Der opnås følgende resultat:

```
x = 4.0000
y = -2.0000
```

Løsningen til ligningssystemet er således: $\underline{\underline{x = 4 \text{ og } y = -2}}$.

(3c)

Der defineres følgende funktioner svarende til højresiderne i ligningssystemet:

$f_1(x,y) = 4x - 5y - y^2 + xy + 10$ og $f_2(x,y) = 6x + y - 15$

Den tilhørende Jacobi-matrix bliver således:

$$J = \begin{pmatrix} \dfrac{\partial f_1(x,y)}{\partial x} & \dfrac{\partial f_1(x,y)}{\partial y} \\ \dfrac{\partial f_2(x,y)}{\partial x} & \dfrac{\partial f_2(x,y)}{\partial y} \end{pmatrix} = \begin{pmatrix} y+4 & x-2y-5 \\ 6 & 1 \end{pmatrix}$$

Derfor bliver den færdige funktion, `Jogf`, som følger:

```
function [J,f] = Jogf(X)
x = X(1); y = X(2);
f = [4*x - 5*y - y^2 + x*y + 10
     6*x + y - 15];
J = [y + 4, x - 2*y - 5
     6,     1          ];
```

(3d)

Ved at anvende MATLAB-kommandoen

```
[X,f,ea,iter] = newtmult(@Jogf,[0;0],1)
```

fås resultatet

```
X =
    2.0000
    3.0001
f =
   -0.2925
         0
ea =
    0.9528
iter =
     3
```

Ifølge dette resultat har ligningssystemet derfor en løsning, hvor $\underline{\underline{x = 2{,}00 \text{ og } y = 3{,}00}}$. Da den approksimative relative fejl er $ea = 0{,}9528\%$, er fejlen under det ønskede niveau på 1%, hvor den ønskede nøjagtighed er opnået.

PS! Hvis den alternative version af Jogf anvendes, giver ovenstående MATLAB-kommando resultaterne `X = [2.0038; 3.0006]`, `f = [-0.0560; -0.0617]`, `ea = 0.6887` og `iter = 6`, altså også en løsning, hvor $x = 2{,}00$ og $y = 3{,}00$.

(3e)
Idet den approksimative relative fejl er 0,0065%, fås følgende tabel:

Ubekendt	Løsningsværdi	Approksimativ absolut fejl	Usikkerhedsinterval
x	$3{,}4524$	$\lvert x * ea/100\rvert = 0{,}0002$	$x \pm 0{,}0002 = [3{,}4522;\ 3{,}4526]$
y	$-5{,}7143$	$\lvert y * ea/100\rvert = 0{,}0004$	$y \pm 0{,}0004 = [-5{,}7147;\ -5{,}7139]$

Vedr. x: Der skal afrundes til to decimaler, for at endepunkterne i usikkerhedsintervaller bliver ens.

Vedr. y: Der skal afrundes til to decimaler, for at endepunkterne i usikkerhedsintervaller bliver ens.

Angivet med sikre decimaler efter afrunding bliver løsningen derfor: $\underline{x = 3{,}45 \text{ og } y = -5{,}71}$.

OPGAVE 4

(4a)
Her følger talresultater og formler i forbindelse med løsning vha. Eulers metode:

	A	B	C	D
1	i	t	y	dy/dt
2	0	0	5	6
3	1	0,5	8	5,179866
4	2	1	10,58993	4,386063
5	3	1,5	12,78296	3,701916
6	4	2	14,63392	3,122903
7	5	2,5	16,19537	2,636471
8	6	3	17,51361	2,22923
9	7	3,5	18,62822	1,888805
10	8	4	19,57263	1,60434
11	9	4,5	20,3748	1,366543
12	10	5	21,05807	1,167566
13	11	5,5	21,64185	1,000838
14	12	6	22,14227	0,860884
15	13	6,5	22,57271	0,743159
16	14	7	22,94429	0,643898
17	15	7,5	23,26624	0,559986
18	16	8	23,54623	0,488849
19	17	8,5	23,79066	0,428359
20	18	9	24,00484	0,376759
21	19	9,5	24,19322	0,332594
22	20	10	24,35951	0,294662

	A	B	C	D
1	i	t	y	dy/dt
2	0	=A2*0,5	5	=0,4*(20+6*(1-EKSP(-0,3*B2^0,8))-C2)
3	1	=A3*0,5	=C2+D2*0,5	=0,4*(20+6*(1-EKSP(-0,3*B3^0,8))-C3)
4	2	=A4*0,5	=C3+D3*0,5	=0,4*(20+6*(1-EKSP(-0,3*B4^0,8))-C4)
5	3	=A5*0,5	=C4+D4*0,5	=0,4*(20+6*(1-EKSP(-0,3*B5^0,8))-C5)
6	4	=A6*0,5	=C5+D5*0,5	=0,4*(20+6*(1-EKSP(-0,3*B6^0,8))-C6)
7	5	=A7*0,5	=C6+D6*0,5	=0,4*(20+6*(1-EKSP(-0,3*B7^0,8))-C7)
8	6	=A8*0,5	=C7+D7*0,5	=0,4*(20+6*(1-EKSP(-0,3*B8^0,8))-C8)
9	7	=A9*0,5	=C8+D8*0,5	=0,4*(20+6*(1-EKSP(-0,3*B9^0,8))-C9)
10	8	=A10*0,5	=C9+D9*0,5	=0,4*(20+6*(1-EKSP(-0,3*B10^0,8))-C10)
11	9	=A11*0,5	=C10+D10*0,5	=0,4*(20+6*(1-EKSP(-0,3*B11^0,8))-C11)
12	10	=A12*0,5	=C11+D11*0,5	=0,4*(20+6*(1-EKSP(-0,3*B12^0,8))-C12)
13	11	=A13*0,5	=C12+D12*0,5	=0,4*(20+6*(1-EKSP(-0,3*B13^0,8))-C13)
14	12	=A14*0,5	=C13+D13*0,5	=0,4*(20+6*(1-EKSP(-0,3*B14^0,8))-C14)
15	13	=A15*0,5	=C14+D14*0,5	=0,4*(20+6*(1-EKSP(-0,3*B15^0,8))-C15)
16	14	=A16*0,5	=C15+D15*0,5	=0,4*(20+6*(1-EKSP(-0,3*B16^0,8))-C16)
17	15	=A17*0,5	=C16+D16*0,5	=0,4*(20+6*(1-EKSP(-0,3*B17^0,8))-C17)
18	16	=A18*0,5	=C17+D17*0,5	=0,4*(20+6*(1-EKSP(-0,3*B18^0,8))-C18)
19	17	=A19*0,5	=C18+D18*0,5	=0,4*(20+6*(1-EKSP(-0,3*B19^0,8))-C19)
20	18	=A20*0,5	=C19+D19*0,5	=0,4*(20+6*(1-EKSP(-0,3*B20^0,8))-C20)
21	19	=A21*0,5	=C20+D20*0,5	=0,4*(20+6*(1-EKSP(-0,3*B21^0,8))-C21)
22	20	=A22*0,5	=C21+D21*0,5	=0,4*(20+6*(1-EKSP(-0,3*B22^0,8))-C22)

Mælketemperaturen efter 5 timer aflæses i række 12 til <u>21,1°C</u>.

(4b)
Her følger MATLAB-kommandoer til gennemførelse af Euler-løsningen:

```
dydt = @(t,y) 0.4*(20+6*(1-exp(-0.3*t^0.8))-y);
tidsinterval = [0 10];
y0 = 5;
```

```
h = 0.25;
[t,y] = eulode(dydt,tidsinterval,y0,h);
disp(' t (timer)  y (gr.C)')
disp([t y])
```

Følgende resultater opnås:

```
 t (timer)  y (gr.C)
         0    5.0000
    0.2500    6.5000
    0.5000    7.9065
    0.7500    9.2108
    1.0000   10.4170
    1.2500   11.5308
    1.5000   12.5585
    1.7500   13.5065
    2.0000   14.3806
    2.2500   15.1867
    2.5000   15.9300
    2.7500   16.6157
    3.0000   17.2483
    3.2500   17.8321
    3.5000   18.3712
    3.7500   18.8691
    4.0000   19.3292
    4.2500   19.7546
    4.5000   20.1482
    4.7500   20.5125
    5.0000   20.8499
    5.2500   21.1626
    5.5000   21.4526
    5.7500   21.7217
    6.0000   21.9717
    6.2500   22.2040
    6.5000   22.4200
    6.7500   22.6210
    7.0000   22.8083
    7.2500   22.9829
    7.5000   23.1458
    7.7500   23.2978
    8.0000   23.4399
    8.2500   23.5727
    8.5000   23.6970
    8.7500   23.8135
    9.0000   23.9227
    9.2500   24.0251
    9.5000   24.1212
    9.7500   24.2116
   10.0000   24.2965
```

Mælketemperaturen efter 5 timer aflæses til 20,8°C.

(4c)
Da Eulers metode er en førsteordensmetode, er afvigelsen fra den korrekte løsning tilnærmelsesvist proportional med tidsskridtet. Derfor vil man ved en halvering af tidsskridtet fra 0,5 time til 0,25 time forvente, at afvigelsen også halveres. Dette efterprøves ved at dividere de to afvigelser med hinanden: $\frac{0{,}0666}{0{,}1296} = 0{,}5139$. Det ses, at afvigelsen ca. halveres, hvorfor reduktionen i afvigelse er af den forventede størrelsesorden.