Kapittel 6.3

Oppgave CP10

Resultat

Plot

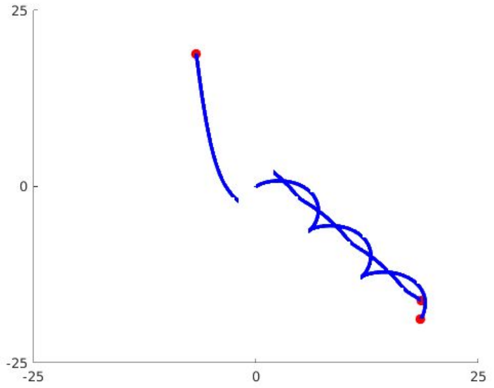

Fig 1. x_1' = 0.2

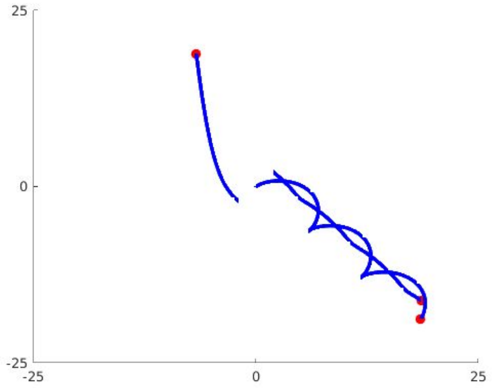

Fig 2. x_1' = 0.20001

Skulle blitt ulike ifølge oppgaveteksten - det fikk ikke vi til å reprodusere.

Kildekode

CP10.m

```
% task a
figure(1)
orbit([0 100], [2, 0.2, 2, -0.2, 0, 0, 0, 0, -2, -0.2, -2, 0.2], 10000, 5)

% task b
figure(2)
orbit([0 100], [2, 0.20001, 2, -0.2, 0, 0, 0, 0, -2, -0.2, -2, 0.2], 10000, 5)
```

orbit.m

```
function orbit(int, ic, n, p)
   % plots n points
    h = (int(2) - int(1)) / n;

    % initial conditions
    x1 = ic(1); vx1 = ic(2);  y1 = ic(3);  vy1 = ic(4);
    x2 = ic(5); vx2 = ic(6);  y2 = ic(7);  vy2 = ic(8);
    x3 = ic(9); vx3 = ic(10); y3 = ic(11); vy3 = ic(12);

    % y-vector
    y(1, :) = [x1 vx1 y1 vy1 x2 vx2 y2 vy2 x3 vx3 y3 vy3];
    t(1) = int(1);

    % sets up plot
    set(gca, 'XLim', [-25 25], 'YLim', [-25 25], 'XTick', [-25 0 25],
       'YTick', ...[-25 0 25], 'Visible', 'on');
    cla;

    % heads
    head1=line('color', 'r', 'Marker', '.', 'markersize', 25, ...
      'xdata', [], 'ydata', []);
    head2=line('color', 'r', 'Marker', '.', 'markersize', 25, ...
      'xdata', [], 'ydata', []);
    head3=line('color', 'r', 'Marker', '.', 'markersize', 25, ...
      'xdata', [], 'ydata', []);

    % tails
    l1 = animatedline('Color', 'b', 'LineWidth', 3);
    l2 = animatedline('Color', 'b', 'LineWidth', 3);
    l3 = animatedline('Color', 'b', 'LineWidth', 3);

    % calculates data
    for k = 1:n/p
      for i = 1:p
        t(i+1) = t(i) + h;
        y(i+1, :) = eulerstep(t(i), y(i, :), h);
      end

      y(1, :) = y(p+1, :);
      t(1) = t(p+1);

      % sets heads
      set(head1,'xdata', y(1, 1), 'ydata', y(1, 3))
      set(head2,'xdata', y(1, 5), 'ydata', y(1, 7))
      set(head3,'xdata', y(1, 9), 'ydata', y(1, 11))

      % sets tails
      addpoints(l1, y(2:p, 1), y(2:p, 3));
```

```
        addpoints(l2, y(2:p, 5), y(2:p, 7));
        addpoints(l3, y(2:p, 9), y(2:p, 11));

        % draws data
        drawnow;
    end
end
```

ydot.m

```
function z = ydot(~, x)
    g = 1;

    m1 = 0.3;
    m2 = 0.03;
    m3 = 0.03;

    mg1 = m1*g;
    mg2 = m2*g;
    mg3 = m3*g;

    px1 = x(1); py1 = x(3);  vx1 = x(2);  vy1 = x(4);
    px2 = x(5); py2 = x(7);  vx2 = x(6);  vy2 = x(8);
    px3 = x(9); py3 = x(11); vx3 = x(10); vy3 = x(12);

    dist1 = sqrt((px2-px1)^2 + (py2-py1)^2);
    dist2 = sqrt((px2-px3)^2 + (py2-py3)^2);
    dist3 = sqrt((px3-px1)^2 + (py3-py1)^2);

    z = zeros(1, 8);

    z(1)  = vx1;
    z(2)  = (mg2*(px2-px1))/(dist1^3)+(mg3*(px3-px1))/(dist3^3);

    z(3)  = vy1;
    z(4)  = (mg2*(py2-py1))/(dist1^3)+(mg3*(py3-py1))/(dist3^3);

    z(5)  = vx2;
    z(6)  = (mg1*(px1-px2))/(dist1^3)+(mg3*(px3-px2))/(dist2^3);

    z(7)  = vy2;
    z(8)  = (mg1*(py1-py2))/(dist1^3)+(mg3*(py3-py2))/(dist2^3);

    z(9)  = vx3;
    z(10) = (mg1*(px1-px3))/(dist3^3)+(mg2*(px2-px3))/(dist2^3);

    z(11) = vy3;
    z(12) = (mg1*(py1-py3))/(dist3^3)+(mg2*(py2-py3))/(dist2^3);
end
```

eulerstep.m

```
function y = eulerstep(t, x, h)
    y = x + h * ydot(t, x);
end
```

Oppgave CP11

Resultat

Plot

Oppgave a

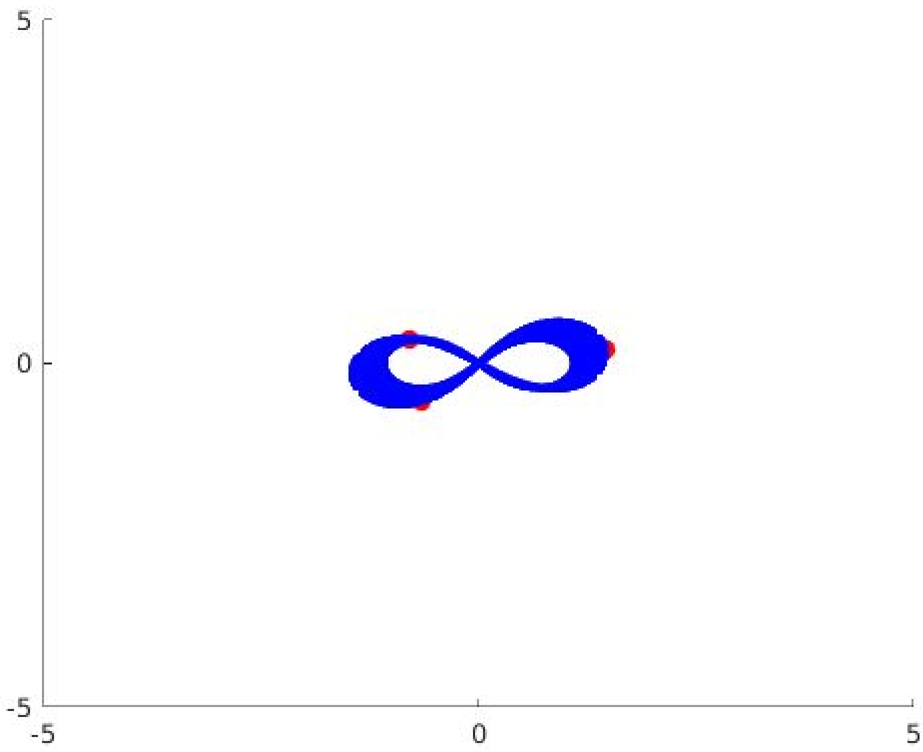

Fig 3.

Oppgave b

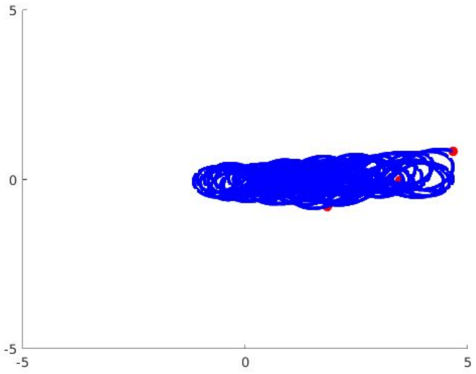

Fig 4. k = 1

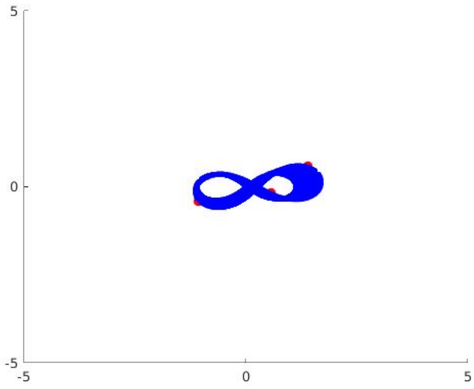

Fig 5. k = 2

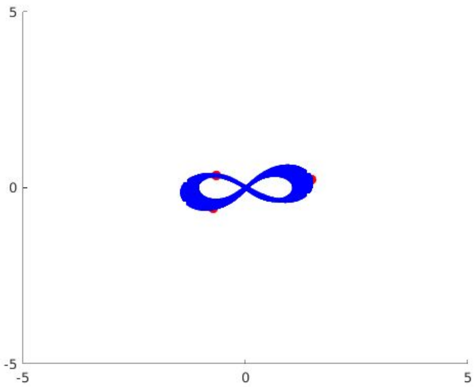

Fig 6. k = 3

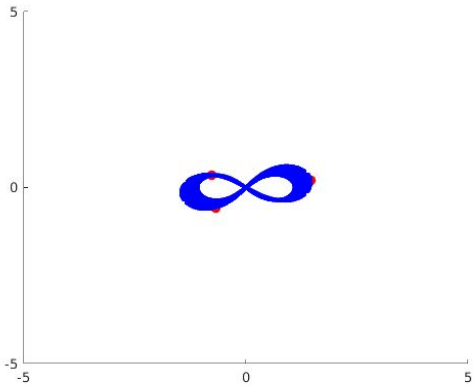

Fig 7. k = 4

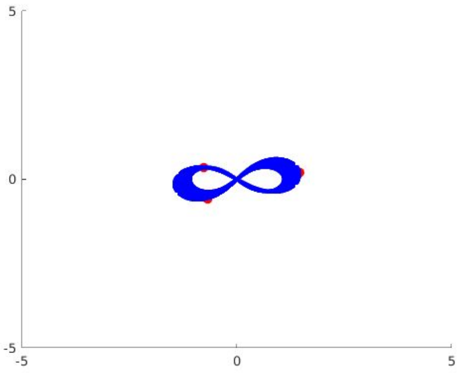

Fig 8. k = 5

Ser at for k <= 1 er mønsteret kaotisk.

Kildekode

CP11.m

```
% task a
figure('Name', 'Task a')
orbit([0 100], [-0.970, -0.466, 0.243, -0.433, 0.970, -0.466, -0.243, -0.433, 0,
0.932, 0, 0.866], 100000, 5)

% task b
for k = 1:5
    figure('Name', 'Task b')
    orbit([0 100], [-0.970, -0.466, 0.243, -0.433, 0.970, -0.466, -0.243,
-0.433, 0, 0.932 + (10^-k), 0, 0.866], 100000, 5)
end
```

orbit.m

```
function orbit(int, ic, n, p)
   % plots n points
    h = (int(2) - int(1)) / n;

    % initial conditons
    x1 = ic(1); vx1 = ic(2);  y1 = ic(3);  vy1 = ic(4);
    x2 = ic(5); vx2 = ic(6);  y2 = ic(7);  vy2 = ic(8);
    x3 = ic(9); vx3 = ic(10); y3 = ic(11); vy3 = ic(12);

    % y-vector
    y(1, :) = [x1 vx1 y1 vy1 x2 vx2 y2 vy2 x3 vx3 y3 vy3];
    t(1) = int(1);

    % sets up plot
    set(gca, 'XLim', [-5 5], 'YLim', [-5 5], 'XTick', [-5 0 5],
       'YTick', ...[-5 0 5], 'Visible', 'on');
    cla;

    % heads
    head1=line('color', 'r', 'Marker', '.', 'markersize', 25, ...
      'xdata', [], 'ydata', []);
    head2=line('color', 'r', 'Marker', '.', 'markersize', 25, ...
      'xdata', [], 'ydata', []);
    head3=line('color', 'r', 'Marker', '.', 'markersize', 25, ...
      'xdata', [], 'ydata', []);

    % tails
    l1 = animatedline('Color', 'b', 'LineWidth', 3);
    l2 = animatedline('Color', 'b', 'LineWidth', 3);
```

```
    l3 = animatedline('Color', 'b', 'LineWidth', 3);

    % calculates data
    for k = 1:n/p
      for i = 1:p
        t(i+1) = t(i) + h;
        y(i+1, :) = eulerstep(t(i), y(i, :), h);
      end

      y(1, :) = y(p+1, :);
      t(1) = t(p+1);

      % sets heads
      set(head1,'xdata', y(1, 1), 'ydata', y(1, 3))
      set(head2,'xdata', y(1, 5), 'ydata', y(1, 7))
      set(head3,'xdata', y(1, 9), 'ydata', y(1, 11))

      % sets tails
      addpoints(l1, y(2:p, 1), y(2:p, 3));
      addpoints(l2, y(2:p, 5), y(2:p, 7));
      addpoints(l3, y(2:p, 9), y(2:p, 11));

      % draws data
      drawnow;
    end
end
```

eulerstep.m

```
function y = eulerstep(t, x, h)
    y = x + h * ydot(t, x);
end
```

ydot.m

```
function z = ydot(~, x)
    g = 1;

    m1 = 1;
    m2 = 1;
    m3 = 1;

    mg1 = m1*g;
    mg2 = m2*g;
    mg3 = m3*g;

    px1 = x(1); py1 = x(3);  vx1 = x(2);  vy1 = x(4);
    px2 = x(5); py2 = x(7);  vx2 = x(6);  vy2 = x(8);
    px3 = x(9); py3 = x(11); vx3 = x(10); vy3 = x(12);

    dist1 = sqrt((px2-px1)^2 + (py2-py1)^2);
    dist2 = sqrt((px2-px3)^2 + (py2-py3)^2);
    dist3 = sqrt((px3-px1)^2 + (py3-py1)^2);

    z = zeros(1, 8);

    z(1) = vx1;
    z(2) = (mg2*(px2-px1))/(dist1^3)+(mg3*(px3-px1))/(dist3^3);

    z(3) = vy1;
    z(4) = (mg2*(py2-py1))/(dist1^3)+(mg3*(py3-py1))/(dist3^3);

    z(5) = vx2;
    z(6) = (mg1*(px1-px2))/(dist1^3)+(mg3*(px3-px2))/(dist2^3);

    z(7) = vy2;
    z(8) = (mg1*(py1-py2))/(dist1^3)+(mg3*(py3-py2))/(dist2^3);

    z(9) = vx3;
    z(10) = (mg1*(px1-px3))/(dist3^3)+(mg2*(px2-px3))/(dist2^3);

    z(11) = vy3;
    z(12) = (mg1*(py1-py3))/(dist3^3)+(mg2*(py2-py3))/(dist2^3);
end
```

Kapittel 6.4

Oppgave CP1b

Resultat

Tabell

t	w	e
0	1	0
0.1	1.00025	8.33888950622086e-05
0.2	1.00250168778125	0.000168537603574048
0.3	1.00877985460098	0.00026076717288781
0.4	1.02119301680933	0.000369498919610134
0.5	1.04203760866844	0.000509296521552294
0.6	1.07395326680144	0.000702077262376477
0.7	1.1201445337832	0.00098123741350653
0.8	1.18469636299413	0.00139895502007503
0.9	1.27302969321169	0.0020389309067661
1	1.39257370568145	0.00303871940464018

Kildekode

CP1b.m

```
[t, w, e] = midpointmethod([0 1], 1, 10);
table(t', w', e', 'VariableNames', {'t', 'w', 'e'})
```

midpointmethod.m

```
function [t, w, e] = midpointmethod(inter, y0, n)
    t(1) = inter(1);
    w(1) = y0;
    e(1) = 0;

    h = (inter(2) - inter(1)) / n;

    for  i = 1:n
        t(i+1) = t(i) + h;
        w(i+1) = midpointstep(t(i), w(i), h);
        e(i+1) = exact(t(i+1)) - w(i+1);
    end
end
```

midtpointstep.m

```
function y = midpointstep(t, w, h)
    y = w + h * ydot(t + (h/2), w + (h/2) * ydot(t, w));
end
```

ydot.m

```
function z = ydot(t, y)
    z = t^2 * y;
end
```

exact.m

```
function y = exact(t)
    y = exp(t^3 / 3);
end
```

Oppgave CP3

Resultat

Tabell [h = 0.1]

t	w	y	e
0	1	1	0
0.1	1.00033338542187	1.00033338889506	3.47318729332358e-09
0.2	1.00267021842184	1.00267022538482	6.96298418922936e-09
0.3	1.00904061125474	1.00904062177387	1.05191246824177e-08
0.4	1.02156250146596	1.02156251572894	1.42629734689592e-08
0.5	1.04254688669115	1.04254690518999	1.84988449003498e-08
0.6	1.07465532004024	1.07465534406381	2.4023574685117e-08
0.7	1.12112573820632	1.12112577119671	3.29903875329052e-08
0.8	1.1860952667244	1.1860953180142	5.12898012949847e-08
0.9	1.27506852922786	1.27506862411846	9.48906042630426e-08
1	1.39561221897344	1.39561242508609	2.06112650946721e-07

Plot [h = 0.1]

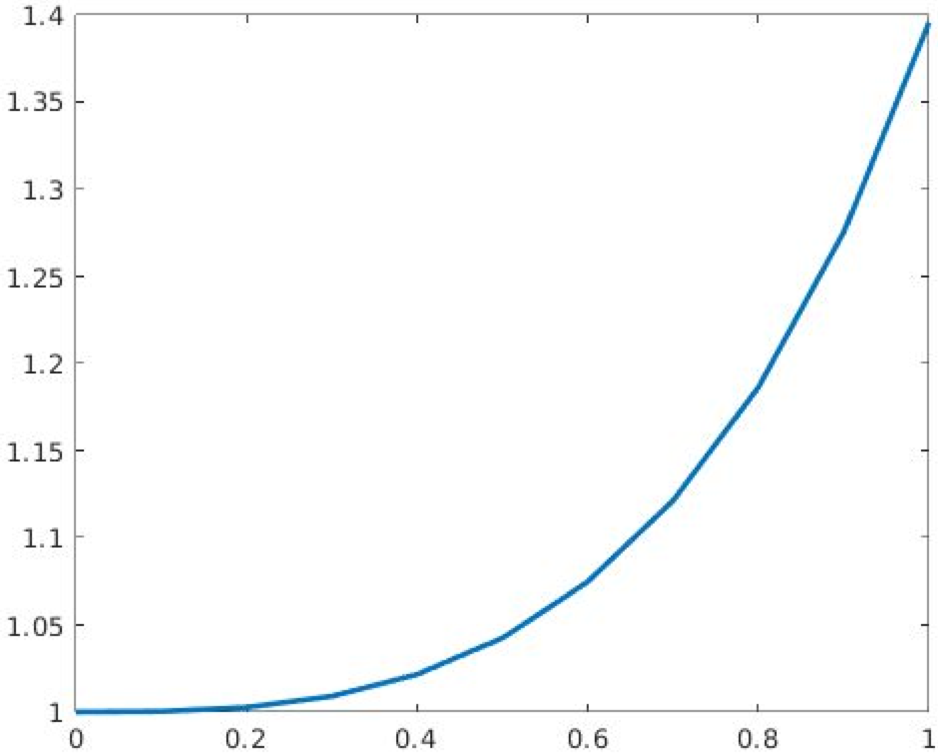

Fig 9. Plot for estimat med h = 0.1

Tabell [h = 0.05]

t	w	y	e
0	1	1	0
0.05	1.00004166748048	1.00004166753473	5.42554889904068e-11
0.1	1.00033338878652	1.00033338889506	1.08542508314713e-10
0.15	1.00112563288693	1.00112563304987	1.6294454674437e-10
0.2	1.00267022516721	1.00267022538482	2.17609485986259e-10
0.25	1.00522192000682	1.0052219202796	2.7277091696476e-10
0.3	1.00904062144508	1.00904062177387	3.28792770787345e-10
0.35	1.01439428040802	1.01439428079427	3.86250142980771e-10
0.4	1.02156251528286	1.02156251572894	4.46077397242561e-10
0.45	1.03084102635681	1.03084102686664	5.09833286699291e-10
0.5	1.04254690460981	1.04254690518999	5.80182346610059e-10
0.55	1.05702497278586	1.05702497344763	6.61766641485428e-10
0.6	1.07465534330105	1.07465534406381	7.62761853678739e-10
0.65	1.09586243577854	1.09586243667615	8.97607321803662e-10
0.7	1.12112577010499	1.12112577119671	1.09171427453703e-09
0.75	1.15099294330175	1.15099294469118	1.38942968241906e-09
0.8	1.18609531614691	1.1860953180142	1.86729387507967e-09
0.85	1.2271670863708	1.22716708902659	2.6557942511829e-09
0.9	1.27506862014383	1.27506862411846	3.97463284507182e-09
0.95	1.33081516701932	1.33081517320866	6.189346812846e-09
1	1.3956124151846	1.39561242508609	9.90149184865174e-09

Plot [h = 0.05]

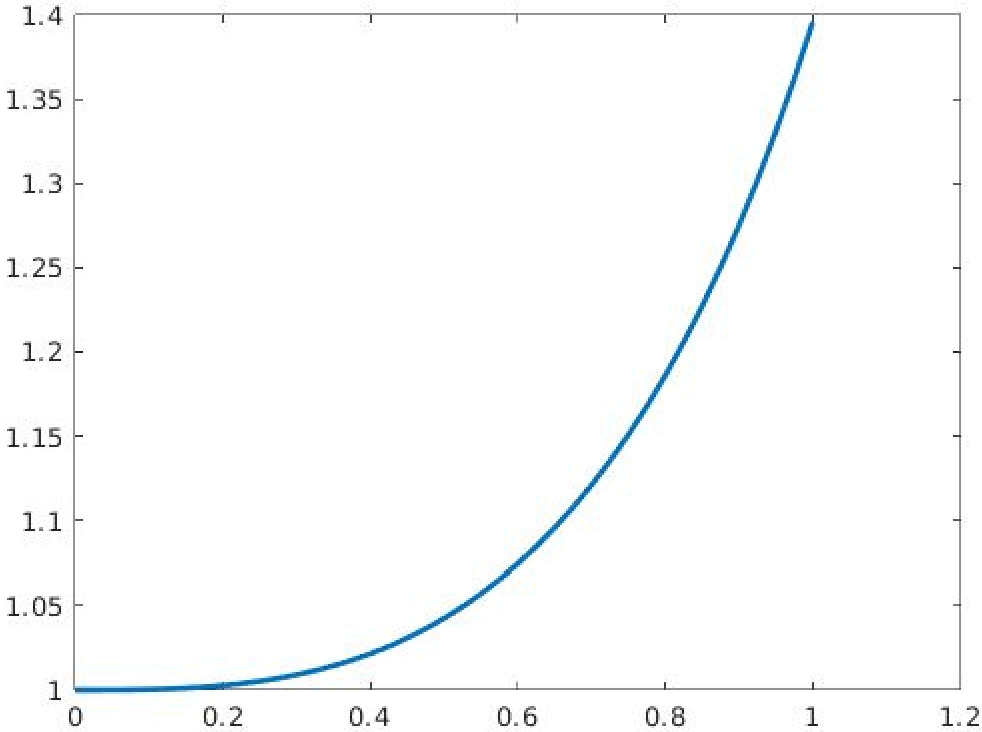

Fig 10. Plot for estimat med h = 0.05

Tabell [h = 0.0025]

t	w	y	e
0	1	1	0
0.025	1.00000520834605	1.0000052083469	8.47544256998845e-13
0.05	1.00004166753304	1.00004166753473	1.69553260320754e-12
0.075	1.00014063488562	1.00014063488816	2.54329890481131e-12
0.1	1.00033338889167	1.00033338889506	3.39195338483478e-12
0.125	1.00065125363605	1.00065125364029	4.24127399867302e-12
0.15	1.00112563304478	1.00112563304987	5.09214892474574e-12
0.175	1.00178805499473	1.00178805500067	5.94502225226279e-12
0.2	1.00267022537802	1.00267022538482	6.80033807043401e-12
0.225	1.00380409225368	1.00380409226134	7.65987273609881e-12
0.25	1.00522192027107	1.0052219202796	8.52429238307195e-12
0.275	1.00695637561131	1.00695637562071	9.39559541279777e-12
0.3	1.00904062176359	1.00904062177387	1.02755581821157e-11
0.325	1.01150842653474	1.01150842654591	1.11668452262847e-11
0.35	1.0143942807822	1.01439428079427	1.20727872143789e-11
0.375	1.01773352946396	1.01773352947696	1.29980470831015e-11
0.4	1.02156251571499	1.02156251572894	1.39483979921806e-11
0.425	1.02591873878922	1.02591873880415	1.49322776366034e-11
0.45	1.03084102685068	1.03084102686664	1.59603441574063e-11
0.475	1.03636972575869	1.03636972577574	1.70476965877242e-11
0.5	1.04254690517178	1.04254690518999	1.82149850758151e-11
0.525	1.04941658349591	1.0494165835154	1.94910754203192e-11
0.55	1.05702497342671	1.05702497344763	2.09150474717035e-11
0.575	1.06542075008668	1.06542075010922	2.25415242027793e-11
0.6	1.07465534403937	1.07465534406381	2.44451126008016e-11
0.625	1.08478326177871	1.08478326180544	2.67272870502211e-11
0.65	1.09586243664662	1.09586243667615	2.95274915629307e-11
0.675	1.10795461353236	1.10795461356539	3.30340199639068e-11
0.7	1.12112577115921	1.12112577119671	3.75008912811836e-11
0.725	1.13544658627473	1.135446586318	4.32702762509507e-11
0.75	1.15099294464038	1.15099294469118	5.07975883579093e-11
0.775	1.16784650437328	1.16784650443396	6.06896755073194e-11
0.8	1.18609531794045	1.1860953180142	7.37485628121703e-11
0.825	1.20583451995726	1.20583452004829	9.10314046365102e-11
0.85	1.22716708891267	1.22716708902659	1.13923537270466e-10
0.875	1.25020469205407	1.2502046921983	1.44234846288782e-10
0.9	1.27506862393414	1.27506862411846	1.84319670637478e-10
0.925	1.30189085057826	1.30189085081549	2.37229569322039e-10
0.950	1.33081517290176	1.33081517320866	3.06906944302909e-10
0.975	1.36199852492776	1.36199852532619	3.98429289560909e-10
1	1.39561242456778	1.39561242508609	5.18314724473612e-10

Plot [h = 0.025]

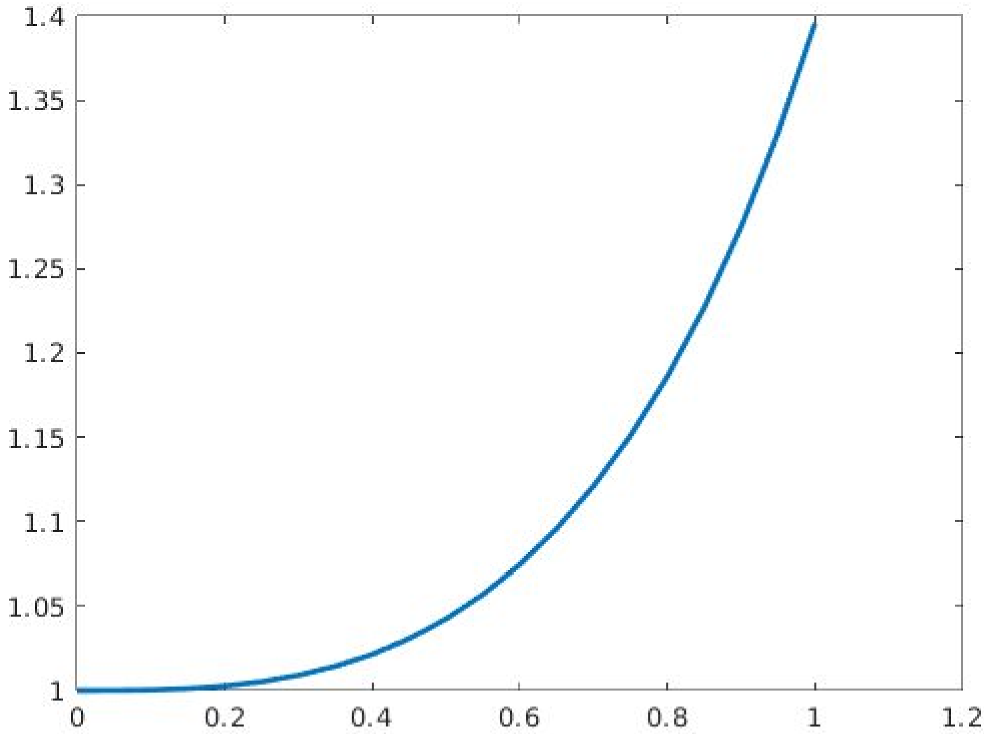

Fig 11. Plot for estimat med h = 0.025

Observasjon

Mindre steglengde gir høyere nøyaktighet på estimatet. Dette er forventet, siden vi “sampler” oftere.

Kildekode

CP3.m

```
% h = 0.1
figure('Name', 'h = 0.1');
[t, w, y, e] = rkmethod([0 1], 1, 10);
table(t', w', y', e', 'VariableNames', {'t', 'w', 'y', 'e'})

% h = 0.05
figure('Name', 'h = 0.05');
[t, w, y, e] = rkmethod([0 1], 1, 20);
table(t', w', y', e', 'VariableNames', {'t', 'w', 'y', 'e'})

% h = 0.025
figure('Name', 'h = 0.025');
[t, w, y, e] = rkmethod([0 1], 1, 40);
table(t', w', y', e', 'VariableNames', {'t', 'w', 'y', 'e'})
```

rkmethod.m

```
function [t, w, y, e] = rkmethod(inter, y0, n)
    t(1) = inter(1);
    w(1) = y0;
    y(1) = y0;
    e(1) = 0;

    h = (inter(2) - inter(1)) / n;

    for  i = 1:n       t(i+1) = t(i) + h;
        w(i+1) = rkstep(t(i), w(i), h);
        y(i+1) = exact(t(i+1));
        e(i+1) = y(i+1) - w(i+1);
    end

    plot(t, w, 'LineWidth', 2);
end
```

rkstep.m

```
function y = rkstep(t, w, h)
    s1 = ydot(t, w);
    s2 = ydot(t + (h/2), w + (h/2)*s1);
    s3 = ydot(t + (h/2), w + (h/2)*s2);
    s4 = ydot(t + h, w + h*s3);

    y = w + (h/6) * (s1 + 2*s2 + 2*s3 + s4);
end
```

ydot.m

```
function z = ydot(t, y)
    z = t^2 * y;
end
```

exact.m

```
function y = exact(t)
    y = exp(t^3 / 3);
end
```

Oppgave CP11

Resultat

Plot

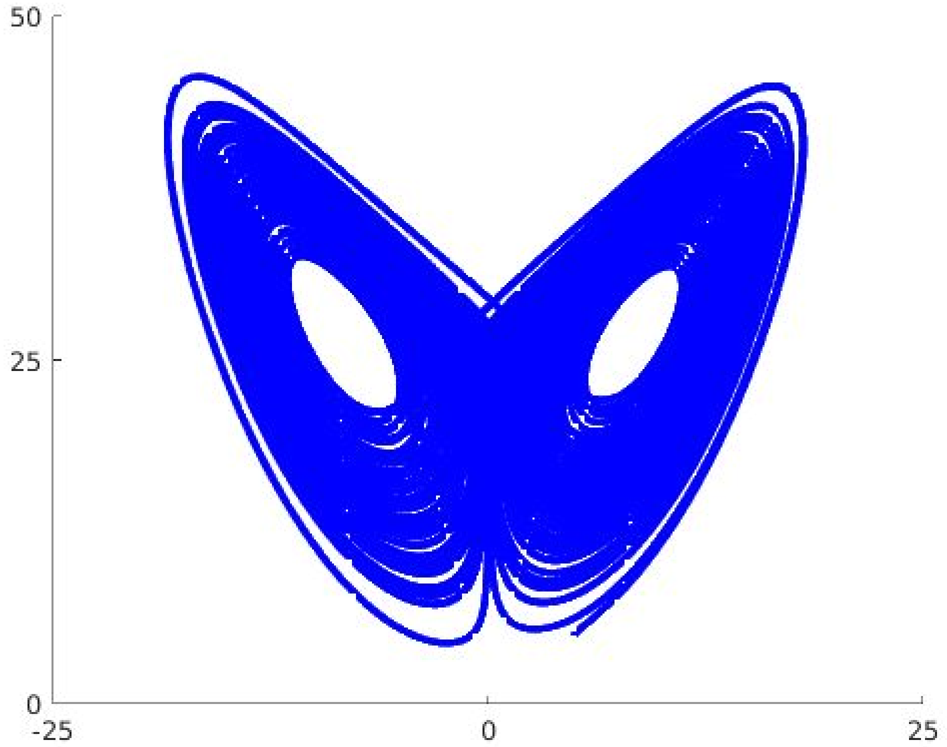

Fig 12. Plot for løsning til Lorenz ligningene v.h.a. Runge Kutta

Kildekode

CP11.m

```
orbit([0 100], [5, 5, 5], 100000, 5)
```

orbit.m

```
function orbit(int, ic, n, p)
   % plots n points
    h = (int(2) - int(1)) / n;

    % initial conditions
    x0 = ic(1); vx0 = ic(2); y0 = ic(3);


    % y-vector
    y(1, :) = [x0 vx0 y0];
    t(1) = int(1);

    % sets up plot
    set(gca, 'XLim', [-25 25], 'YLim', [0 50], 'XTick', [-25 0 25], 'YTick', ...
      [0 25 50], 'Visible', 'on');
    cla;

    % heads
    head = line('color', 'r', 'Marker', '.', 'markersize', 25, ...
      'xdata', [], 'ydata', []);

    % tails
    l = animatedline('Color', 'b', 'LineWidth', 3);

    % calculates data
    for k = 1:n/p
      for i = 1:p
        t(i+1) = t(i) + h;
        y(i+1, :) = rkstep(t(i), y(i, :), h);
      end

      y(1, :) = y(p+1, :);
      t(1) = t(p+1);

      % sets head
      set(head,'xdata', y(1, 1), 'ydata', y(1, 3))

      % sets tail
      addpoints(l, y(2:p, 1), y(2:p, 3));

      % draws data
      drawnow;
    end
end
```

rkstep.m

```
function y = rkstep(t, w, h)
    s1 = ydot(t, w);
    s2 = ydot(t + (h/2), w + (h/2)*s1);
    s3 = ydot(t + (h/2), w + (h/2)*s2);
    s4 = ydot(t + h, w + h*s3);

    y = w + (h/6) * (s1 + 2*s2 + 2*s3 + s4);
end
```

ydot.m

```
function z = ydot(~, y)
  s = 10;
  r = 28;
  b = 8/3;

  z(1) = -s*y(1) + s*y(2);
  z(2) = -y(1)*y(3) + r*y(1) - y(2);
  z(3) = y(1)*y(2) - b*y(3);
end
```