

Bjarki Geir Benediktsson,
Haukur Óskar Þorgeirsson,
Matthías Páll Gissurarson

TÖLULEG GREINING
HEIMAVERKEFNI 3
12. APRÍL 2013

Töluleg Greining

Heimaverkefni 3

Bjarki Geir Benediktsson, Haukur Óskar Þorgeirsson, Matthías Páll Gissurarson
Kennari: Máni Maríus Viðarsson

12. apríl 2013

1 Forsagnar- og Leiðréttingaraðferð

1.1

```
1 function [wi, ti] = adams_pc5 ( RHS, t0, x0, tf, N )
2 %%adams_pc5
3 %ADAMS_PC5 approximate the solution of the initial value problem
4 %
5 %           x'(t) = RHS( t, x ),      x(t0) = x0
6 %
7 %       using the Adams fifth-order predictor / corrector scheme
8 %       - this routine will work for a system of first-order
9 %       equations as well as for a single equation
10 %
11 %       the classical fifth-order Runge-Kutta method is used to
12 %       initialize the predictor / corrector scheme
13 %
14 %
15 %       calling sequences:
16 %       [wi, ti] = adams_pc5 ( RHS, t0, x0, tf, N )
17 %       adams_pc5 ( RHS, t0, x0, tf, N )
18 %
19 %       inputs:
20 %       RHS      string containing name of m-file defining the
21 %                right-hand side of the differential equation; the
22 %                m-file must take two inputs - first, the value of
23 %                the independent variable; second, the value of the
24 %                dependent variable
25 %       t0       initial value of the independent variable
26 %       x0       initial value of the dependent variable(s)
27 %                if solving a system of equations, this should be a
28 %                row vector containing all initial values
29 %       tf       final value of the independent variable
30 %       N        number of uniformly sized time steps to be taken to
31 %                advance the solution from t = t0 to t = tf
32 %
33 %       output:
34 %       wi       vector / matrix containing values of the approximate
35 %                solution to the differential equation
36 %       ti       vector containing the values of the independent
37 %                variable at which an approximate solution has been
38 %                obtained
39 %
40
41 neqn = length ( x0 );
42 ti = linspace ( t0, tf, N+1 );
43 wi = [ zeros( neqn, N+1 ) ];
44 wi(1:neqn, 1) = x0';
```

```

45
46 h = ( tf - t0 ) / N;
47 oldf = zeros(3,neqn);
48
49 %
50 % generate starting values using classical 4th order RK method
51 % remember to save function values
52 %
53
54 for i = 1:4
55     oldf(i,1:neqn) = feval ( RHS, t0, x0 );
56     k1 = h * oldf(i,:);
57     k2 = h * feval ( RHS, t0 + h/4, x0 + k1/4 );
58     k3 = h * feval ( RHS, t0 + 3*h/8, x0 + k1*3/32 + k2*9/32 );
59     k4 = h * feval ( RHS, t0 + 12/13*h, x0 + 1932/2197*k1 + 7200/2197*k2 + 7296/2197*k3 );
60     k5 = h * feval ( RHS, t0 + h, x0 + 439/216*k1 - 8*k2 + 3680/513*k3 + 845/4104*k4 );
61     k6 = h * feval ( RHS, t0 + h/2, x0 - 8/27*k1 + 2*k2 - 3544/2565*k3 + 1859/4104*k4 - ...
        11/40*k5 );
62     x0 = x0 + 16/135*k1 + 6656/12825*k3 + 28561/56430*k4 - 9/50*k5 + 2/55*k6;
63     t0 = t0 + h;
64
65     wi(1:neqn,i+1) = x0';
66 end;
67
68 %
69 % continue time stepping with 5th order Adams Predictor / Corrector
70 %
71
72 for i = 4:N
73     fnew = feval ( RHS, t0, x0 );
74     xtilde = x0 + h*( 1901/720*fnew - 1387/360*oldf(4,:) + 109/30*oldf(3,:) - ...
        637/360*oldf(2,:) + 251/720*oldf(1,:));
75     fnew1 = feval ( RHS, t0+h, xtilde );
76     x0 = x0 + h/720*(251*fnew1 + 646*fnew - 264*oldf(4,:) + 106*oldf(3,:) - 19*oldf(2,:));
77     oldf(1,1:neqn) = oldf(2,1:neqn);
78     oldf(2,1:neqn) = oldf(3,1:neqn);
79     oldf(3,1:neqn) = oldf(4,1:neqn);
80     oldf(4,1:neqn) = fnew;
81     t0 = t0 + h;
82
83     wi(1:neqn,i+1) = x0';
84 end;

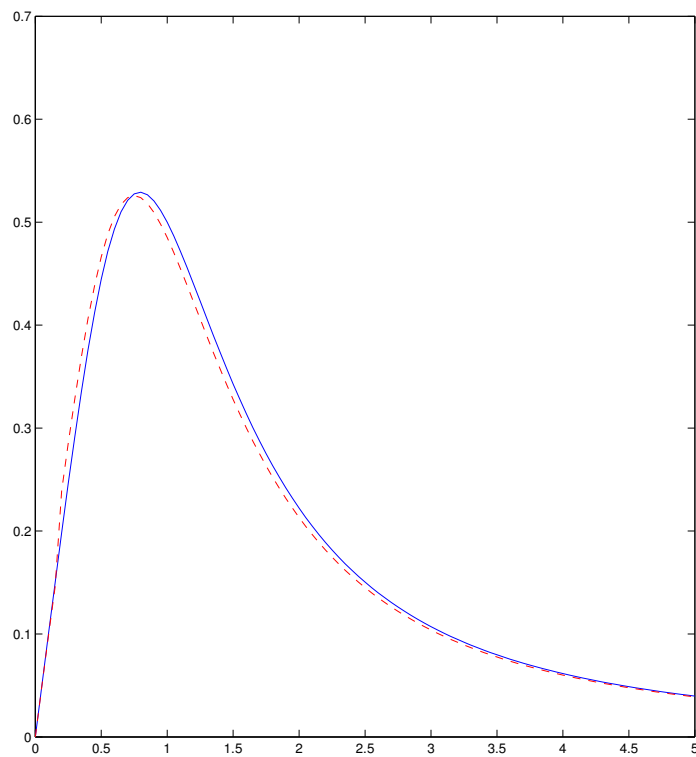
```

1.2

```

1 function [adms, err] = adams_pc5plot (N)
2 %%adams_pc5plot
3 %A function that approximates x'(t) =
4 %RHS = @(t,x) -3*t*x^2 + 1/(1+t^3)
5 %And plots the real function in addition to the approximation, and
6 %a graph of the error
7 RHS = @(t,x) -3*t*x^2 + 1/(1+t^3)
8 t = linspace(0,5,N+1);
9 x = @(t) t/(1+t^3);
10 y = arrayfun(x,t);
11 plot(t,y)
12 hold on
13 adms = adams_pc5(RHS,0,0,5,N)
14 diff = @(x,y) abs(x-y)
15 err = arrayfun(diff,adms,y)
16 plot(t, adms,'r—')
17 figure()
18 plot(t,err)

```



Mynd 1: Fallið sem prófa átti í 1, auk nálgun þess með adams_pc5 sem rautt brotastrík

```

1 %%adams_pc5test
2 %a function that test the adams_pc5 function
3 %that approximates x'(t) =
4 %RHS = @(t,x) -3*t*x^2 + 1/(1+t^3)
5 adams_pc5('RHS', 0, 0, 5, 100)

```

Sjá myndir 1 og 2.

1.3

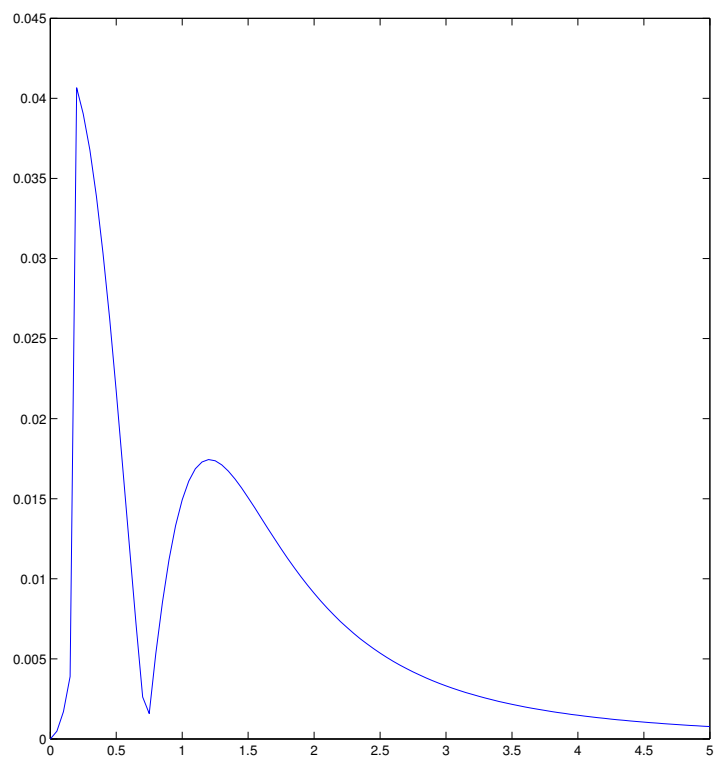
til þess að fá heildarskekkju minni en 10^{-4} þurfti $h = \frac{1}{50000} = 2 * 10^{-5}$

1.4 Tímamælingar

```

1 numtests=100
2 adamstimes=[];
3 rkf45times=[];
4 rk45times=[];
5
6 for i=1:numtests
7     tic
8     adams_pc5('Test',0,0,5,100);
9     adamstimes(i)=toc;

```



Mynd 2: Skekkjan milli nálgunar þess með adams_pc5 og fallsins sem prófa átti í 1 sem fall af t

```

10     tic
11     rkf45('Test',0,0,5,[0.0001,1,0.0001]);
12     rkf45times(i)=toc;
13     tic
14     rkf45('Test',0,0,5,[0.00001,1,0.00001]);
15     rkf45times(i)=toc;
16     tic
17     rk56('Test',0,0,5,[0.00001,1,0.00001]);
18     rk56times(i)=toc;
19 end
20 adamsmeantime=sum(adamstimes)/numtests
21 rkf45meantime=sum(rkf45times)/numtests
22 rk56meantime=sum(rk56times)/numtests

```

Keyrsla á þessu forriti gaf eftirfarandi niðurstöður

fall	meðatími
adams_pc5	0,0082
rkf45	0,0024
rk56	0,0027

2 Einfaldur pendúll

Við fáum að þar sem

$$\theta''(t) + \frac{g}{l} \sin \theta(t) \Leftrightarrow \theta''(t) = -\frac{g}{l} \cdot \sin \theta(t)$$

þá má rita

$$\begin{cases} y(t) = \theta'(t) \\ y'(t) = \theta''(t) = -\frac{g}{l} \cdot \sin \theta(t) \end{cases}$$

Notum þetta til að skilgreina eftirfarandi fall:

```
1 function y = pendulODE (t,x)
2 %%pendulODE
3 %Fall sem er jofnuhneppi fyrir nalgun a einfoldum pendul. tekur inn
4 %t sem er timaskref og x sem eru nuverandi gildi theta og theta'
5     g = 1;
6     l = 1;
7     omega = sqrt(g/l);
8     y(1) = x(2);
9     y(2) = -(omega^2)*sin(x(1));
```

```
1 function y = pendulApprox(x_0)
2 %%pendulApprox
3 %Forrit sem gerir nalgun a pendul fra 0 upp i 5 med upphafsgildin
4 %i x_0, [theta0,theta1]
5     y = adams_pc5('pendulODE',0,x_0,5,100);
```

Við notuðum síðan þetta til að teikna hreyfimyndir, en það var gert með

```
1 function y = pendull(RHS,lotur,n,theta0,theta1,omega,t0,res)
2 %% pendull.m
3 % Skipanaskra sem byr til hreyfimynd af einfoldum penduli, tekur
4 % inn ODE pendulsins sem RHS, hve margar lotur, hve margar myndir per lotu, upphafsgildin
5 % theta0,theta1, omega og t0, og hve margraltfleiri itranir af
6 % nalgun eru en rommum.
7 aviobj = avifile('pendull.avi','compression','None','fps',16); %#ok<REMF1>
8 % Segir til um nafn myndbandsins, thjoppun og fjolda ramma a sek.
9 % Windows notendur aettu ad breyta 'None' i 'Indeo5' eda i einhvern annan
10 % compression moguleika.
11 % Mac, Linux, BSD og onnur styrikerfi thurfa ad thjappa myndbandid
12 % handvirkt utan matlab, t.d med ffmpeg.
13 fig=figure;
14 %% Fastar
15 %lotur = 5;           %Fjoldi lota til ad reikna
16 %n = 25;             %Fjoldi mynda i hverri lotu
17 %theta0 = 2;         %theta(t0)
18 %theta1 = 0;         %theta'(t0)
19 %omega = 1;          %Hornhradi
20 %t0 = 0;
21 %res = 6;
22 %% Jofnurnar
23 theta = @(t) theta0*cos(omega*(t-t0)) + (theta1/omega) * sin(omega*(t-t0));
24 dtheta = @(t) -omega*theta0*sin(omega*(t-t0))+ theta1*cos(omega*(t- ...
25                                     t0));
26
27 %simple = adams_pc5('pendulODE',t0,[theta0,theta1],2*pi*lotur,res*lotur*n);
28 simple = adams_pc5(RHS,t0,[theta0,theta1],2*pi*lotur,res*lotur*n);
29 y = (max(simple(1,:))^2 + max(simple(2,:))^2)
30 thetasimple = @(t) simple(1, res*floor(t*n/(2*pi)) + 1);
31 dthetasimple = @(t) simple(2, res*floor(t*n/(2*pi)) + 1);
32 %%
33 % Thar sem vid hreinsum myndina i hverju skrefi ta thurfum vid ad geyma
34 % fasahnitin i fylki
35 fasahnit = [theta(0);dtheta(0)];
36 simplefasahnit =[thetasimple(0);dthetasimple(0)];
37 for t = 0:2*pi/n:2*pi*lotur
38     %% Fasaritid
39     subplot(2,2,1) %Skipar matlab ad nota seinni hlutan af myndflotinum
```

```

40     fasahnit = [fasahnit(1,:) theta(t); fasahnit(2,:) dtheta(t)]; %Baetir nyju ...
        fasahnitunum vid thau gomlu.
41     simplefasahnit = [simplefasahnit(1,:) thetasimple(t); simplefasahnit(2,:) ...
        dthetasimple(t)]; %Baetir nyju fasahnitunum vid thau gomlu.
42     plot(theta(t),dtheta(t),'ob', 'MarkerSize', 6) %Punkturinn i
43         %fasaritinu
44     hold on % Thurfum ad setja "hold on" her svo vid yfirskrifum ekki linuna i fasaritnu ...
        bara med punktinu
45     plot(fasahnit(1,:),fasahnit(2:),'b') % Linan i fasaritinu
46     hold on
47     plot(thetasimple(t),dthetasimple(t),'or', 'MarkerSize', 6)
48     axis([-theta0-0.2,theta0+0.2,-theta0-0.2,theta0+0.2])
49     axis square
50     hold off
51     subplot(2,2,2) %Skipar matlab ad nota seinni hlutan af myndflotinum
52     plot(simplefasahnit(1,:),simplefasahnit(2:),'r') % Linan i fasaritinu
53     hold on
54     plot(thetasimple(t),dthetasimple(t),'or', 'MarkerSize', 6)
55     %Punkturinn i fasaritinu
56     hold on
57     plot(theta(t),dtheta(t),'ob', 'MarkerSize', 6) %Punkturinn i
58     axis([-theta0-0.2,theta0+0.2,-theta0-0.2,theta0+0.2])
59     axis square
60     hold off
61     %% Pendullinn
62     %Skiptir myndaflothinum i 2x1 fylki og segir matlab ad nota fyrsta stakid
63     % Teikniskipun fyrir pendulinn: teiknum linu fra [0,0] (festipunktur)
64     % i [sin(theta(t)),-cos(theta(t))] sem er stadsetning lodsins
65     % '-o' segir ad vid aetlum ad teikna linu med hringlaga endapunkta
66     % 'MarkerSize' setur staerd endapunktanna
67     % 'MarkerFaceColor' akvadar lit endapunktana
68     subplot(2,2,3)
69     plot([0,sin(theta(t))],[0,-cos(theta(t))],'-o','MarkerSize',8, ...
70         'MarkerFaceColor','b')
71     axis([-1.2,1.2,-1.2,1.2]) %Festir asana
72     axis square %Thvingar matlab til ad hafa x og y asinn jafn
73         %langan
74     hold off
75     subplot(2,2,4)
76     plot([0,sin(thetasimple(t))],[0,-cos(thetasimple(t))],'-o','MarkerSize',8,'MarkerFaceColor','r')
77     axis([-1.2,1.2,-1.2,1.2]) %Festir asana
78     axis square %Thvingar matlab til ad hafa x og y asinn jafn
79         %langan
80     hold off
81     %% Hreyfimynd
82     F = getframe(fig); %Naer i nyjasta ramman
83     aviobj = addframe(aviobj,F); % Og skeytir thvi vid restina
84 end
85 %% Frangangur
86 close(fig); %Lokar myndinni til ad ekki se haegt ad yfirskrifa hana
87 aviobj = close(aviobj); %Lokar og byr til myndbandid

```

Með því að prófa nokkur gildi og horfa á útkomuna, þá komumst við að því að upphafhornið $\theta_0 = 0.42$ gaf okkur frekar góða nálgun að 3 umferð, en eftir það fóru lausnirnar að greinast í sundur.

Til þess að svo gera hreyfimyndina þar sem útslagið er stórt var notaður eftirfarandi forritsbútur:

```

1 %%hr2
2 %fall sem keyrir pendul og byr til hreyfimynd med miklu utslagi
3 pendull('pendulODE',5,25,1,0,1,0,6)

```

3 Róla

Við athugum að jafnan er eins og í 2, nema nú er lengdin auk þess fall af t . Því fæst jöfnuhneppið

$$\begin{cases} y(t) = \theta'(t) \\ y'(t) = \theta''(t) = -\frac{g}{s(t)} \cdot \sin \theta(t) \end{cases}$$

en

$$s(t) = l + a * (\cos \omega t + \varphi),$$

þar sem l er lengdin, a er ústlagið, ω er horntíðnin og φ er fasahliðrunin.

Eftirfarandi tvö föll voru gerð til þess að nota með `adams_pc5`, en fyrra er fyrir rólu sem er að hægja á sér og seinna er fyrir rólu sem er að hámarka ústlagið. Munurinn liggur í mismunandi fasahliðrunum.

```
1 function y = roluODEmin (t,x)
2     g = 1;
3     l = 1;
4     utslag = 0.8;
5     horntidni = sqrt(g/l)/(2*pi*(1+ (x(2)^2)/16 + (x(2)^4)*11/3072));
6     fasahlidrun = 0;
7     s = @(t) l + utslag*cos(horntidni * t + fasahlidrun);
8
9     y(1) = x(2);
10    y(2) = -(g/s(t))*sin(x(1));
11
12 end
```

```
1 function y = roluODEmax (t,x)
2     g = 1;
3     l = 1;
4     utslag = 0.8;
5     horntidni = sqrt(g/l)/(2*pi*(1+ (x(2)^2)/16 + (x(2)^4)*11/3072));
6     fasahlidrun = pi;
7     s = @(t) l + utslag*cos(horntidni * t + fasahlidrun);
8
9     y(1) = x(2);
10    y(2) = -(g/s(t))*sin(x(1));
11
12 end
```

Til þess svo að teikna meðfylgjandi myndbönd var notast við eftirfarandi skrá og `pendull.m` úr fyrri lið.

```
1 pendull('roluODEmin',10, 25, 1,0,1,0,10)
2 pendull('roluODEmax',10, 25, 1,0,1,0,10)
```

Ústlagið örvaðist mest þegar $\omega = 1/T$ eins og gefið var í 3 og fasahornið var π , en ef $\varphi = 0$, þá hægði maður á rólunni. Munurinn liggur í hvenær maður stýttir og lengir bandið, en ef fasahornið er 0, þá er mesta lengingin í miðjunni og minnst í endapunktunum, en öfugt er farið þegar hægt er á rólunni. Útskýringin á þessu er að maður er að nýta sér hröðunina sem þyngdarkrafturinn veitir, en með því að vera að lengja á meðan maður dettur er maður í raun að detta meira beint niður.

4 Kúluþendúll

Jöfnuhneppið er einfaldlega það sama og gefið er í liðnum, en það má sjá í

```
1 function y = kulupendulODE(t,x)
2     m = 1;
```



```

3      l = 1;
4      g = 9.82;
5      om = 1/(m*l*1);
6
7      y(1) = om*x(3);
8      y(2) = om*x(4)/(sin(x(1))^2);
9      y(3) = om*x(4)^2*(cos(x(1))/(sin(x(1))^3)) - m*l*g*sin(x(1));
10     y(4) = 0;

```

Eftirfarandi forritsbútur var notaður til að búa til myndbandið.

```

1 pendull3d('kulupendulODE',10,50,[1,0.5,0,0.4],0,10)

```

Það var gert með hjálp eftirfarandi falls, sem er pendúlsteiknifallið breytt til þess að ráða við þrívídd.

```

1 function y = pendull3d(RHS,lotur,n,x0,t0,res)
2 %% pendull.m
3 % Skipanaskra sem byr til hreyfimynd af einföldum penduli, tekur
4 % inn ODE pendulsins sem RHS, hve margar lotur, hve margar myndir per lotu, upphafsgildin
5 % theta0,theta1, omega og t0, og hve margfaltfleiri itranir af
6 % nalgun eru en rommum.
7 aviobj = avifile('pendull.avi','compression','None','fps',16); %#ok<REMF1>
8 % Segir til um nafn myndbandsins, thjoppun og fjolda ramma a sek.
9 % Windows notendur aettu ad breyta 'None' i 'Indeo5' eda i einhvern annan
10 % compression moguleika.
11 % Mac, Linux, BSD og onnur styrikerfi thurfa ad thjappa myndbandid
12 % handvirkir utan matlab, t.d med ffmpeg.
13 fig=figure;
14 %% Fastar
15 %lotur = 5;           %Fjoldi lota til ad reikna
16 %n = 25;             %Fjoldi mynda i hverri lotu
17 %theta0 = 2;         %theta(t0)
18 %theta1 = 0;         %theta'(t0)
19 %omega = 1;          %Hornhradi
20 %t0 = 0;
21 %res = 6;
22 %% Jofnurnar
23
24 simple = adams_pc5(RHS,t0,x0,2*pi*lotur,res*lotur*n);
25 theta = @(t) simple(1, res*floor(t*n/(2*pi)) + 1);
26 phi = @(t) simple(2, res*floor(t*n/(2*pi)) + 1);
27 ptheta = @(t) simple(3, res*floor(t*n/(2*pi)) + 1);
28 pphi = @(t) simple(4, res*floor(t*n/(2*pi)) + 1);
29 %%
30 % Thar sem vid hreinsum myndina i hverju skrefi ta thurfum vid ad geyma
31 % fasahnitin i fylki
32 %fasahnit = [theta(0);dtheta(0)];
33 fasahnit = [theta(0);phi(0)];
34 x = @(t) sin(theta(t))*cos(phi(t));
35 y = @(t) sin(theta(t))*sin(phi(t));
36 z = @(t) -cos(theta(t)); %theta er midad vid ofugan z-as en venjulega
37 for t = 0:2*pi/n:2*pi*lotur
38     %% Fasaritid
39     %subplot(2,1,1) %Skipar matlab ad nota seinni hlutan af myndflotinum
40
41     fasahnit = [fasahnit(1,:) x(t); fasahnit(2,:) y(t)]; %Baetir nyju fasahnitunum vid ...
42     %thau gomlu.
43
44     %plot(x(t),y(t),'ob', 'MarkerSize', 6) %Punkturinn i
45     %fasaritinu
46
47     %hold on % Thurfum ad setja "hold on" her svo vid yfirskrifum ekki linuna i fasaritnu ...
48     %bara med punktinum
49
50     %plot(fasahnit(1,:),fasahnit(2:,:), 'b') % Linan i fasaritinu
51     %hold on

```

```

50     %axis([-1.2,1.2,-1.2,1.2])
51     %axis square
52
53     %hold off
54     %% Pendullinn
55     %Skiptir myndaflothinum í 2x1 fylki og segir matlab að nota fyrsta stakid
56     % Teiknistíðun fyrir pendulinn: teiknum línu frá [0,0] (festipunktur)
57     % í [sin(theta(t)), -cos(theta(t))] sem er staðsetning lóðsins
58     % '-' segir að við ætlum að teikna línu með hringlaga endapunkta
59     % 'MarkerSize' setur stærð endapunktanna
60     % 'MarkerFaceColor' ákvarðar lit endapunktana
61
62
63     %subplot(2,1,2)
64     plot3([0,x(t)], [0,y(t)], [0,z(t)], '-o', 'MarkerSize', 8, 'MarkerFaceColor', 'b')
65     hold on
66     plot3(fasahnit(1,:), fasahnit(2,:), zeros(1,length(fasahnit(2,:))-1), 'r', ...
67           'MarkerSize', 8, 'MarkerFaceColor', 'r')
68     % hold off
69     grid %minor
70     axis([-1.2,1.2,-1.2,1.2,-1.2,1.2]) %Festir ásana
71     axis square %Þvingar matlab til að hafa x og y ásinn jafn
72     hold off
73     %% Hreyfimynd
74     F = getframe(fig); %Naer í nýjasta ramman
75     aviobj = addframe(aviobj,F); % Og skeytir því við restina
76 end
77 %% Frangangur
78 close(fig); %Lokar myndinni til að ekki sé hægt að yfirskrifa hana
79 aviobj = close(aviobj); %Lokar og byr til myndbandid

```

5 Lorenz

Að skýrslunni unnu : _____