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

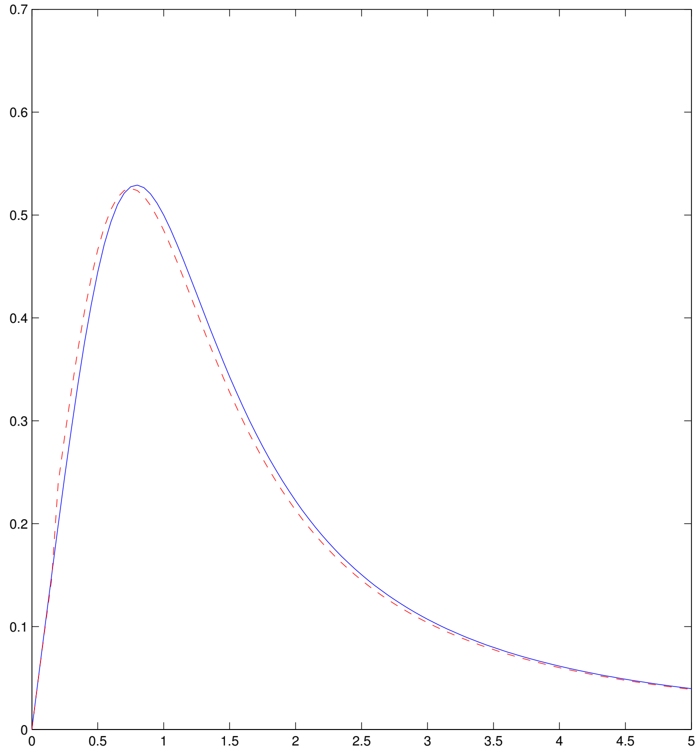
1.1

```
function [wi, ti] = adams_pc5 ( RHS, t0, x0, tf, N )
%%adams_pc5
%ADAMS_PC5  approximate the solution of the initial value problem
%
%                       x'(t) = RHS( t, x ),    x(t0) = x0
%
%           using the Adams fifth-order predictor / corrector scheme
%           - this routine will work for a system of first-order
%           equations as well as for a single equation
%
%           the classical fifth-order Runge-Kutta method is used to
%           initialize the predictor / corrector scheme
%
%
%     calling sequences:
%             [wi, ti] = adams_pc5 ( RHS, t0, x0, tf, N )
%             adams_pc5 ( RHS, t0, x0, tf, N )
%
%     inputs:
%             RHS     string containing name of m-file defining the
%                     right-hand side of the differential equation;  the
%                     m-file must take two inputs - first, the value of
%                     the independent variable; second, the value of the
%                     dependent variable
%             t0      initial value of the independent variable
%             x0      initial value of the dependent variable(s)
%                     if solving a system of equations, this should be a
%                     row vector containing all initial values
%             tf      final value of the independent variable
%             N       number of uniformly sized time steps to be taken to
%                     advance the solution from t = t0 to t = tf
%
%     output:
%             wi      vector / matrix containing values of the approximate
%                     solution to the differential equation
%             ti      vector containing the values of the independent
%                     variable at which an approximate solution has been
%                     obtained
%

neqn = length ( x0 );
ti = linspace ( t0, tf, N+1 );
wi = [ zeros( neqn, N+1 ) ];
wi(1:neqn, 1) = x0';
```

```matlab

h = ( tf - t0 ) / N;
oldf = zeros(3,neqn);

%
%  generate starting values using classical 4th order RK method
%  remember to save function values
%

for i = 1:4
    oldf(i,1:neqn) = feval ( RHS, t0, x0 );
    k1 = h * oldf(i,:);
    k2 = h * feval ( RHS, t0 + h/4, x0 + k1/4 );
    k3 = h * feval ( RHS, t0 + 3*h/8, x0 + k1*3/32 + k2*9/32 );
    k4 = h * feval ( RHS, t0 + 12/13*h, x0 + 1932/2197*k1 + 7200/2197*k2 + 7296/2197*k3 );
    k5 = h * feval ( RHS, t0 + h, x0 + 439/216*k1 - 8*k2 + 3680/513*k3 + 845/4104*k4);
    k6 = h * feval ( RHS, t0 + h/2, x0 - 8/27*k1 + 2*k2 - 3544/2565*k3 + 1859/4104*k4 - ...
        11/40*k5);
    x0 = x0 + 16/135*k1 + 6656/12825*k3 + 28561/56430*k4 - 9/50*k5 + 2/55*k6;
    t0 = t0 + h;

    wi(1:neqn,i+1) = x0';
end;

%
%  continue time stepping with 5th order Adams Predictor / Corrector
%

for i = 4:N
    fnew = feval ( RHS, t0, x0 );
    xtilde = x0 + h*( 1901/720*fnew - 1387/360*oldf(4,:) + 109/30*oldf(3,:) - ...
        637/360*oldf(2,:) + 251/720*oldf(1,:));
    fnew1 = feval ( RHS, t0+h, xtilde );
    x0 = x0 + h/720*(251*fnew1 + 646*fnew - 264*oldf(4,:) + 106*oldf(3,:) - 19*oldf(2,:));
    oldf(1,1:neqn) = oldf(2,1:neqn);
    oldf(2,1:neqn) = oldf(3,1:neqn);
    oldf(3,1:neqn) = oldf(4,1:neqn);
    oldf(4,1:neqn) = fnew;
    t0 = t0 + h;

    wi(1:neqn,i+1) = x0';
end;
```

1.2

```matlab
function [adms, err] = adams_pc5plot (N)
%%adams_pc5plot
%A function that approximates x'(t) =
%RHS = @(t,x) -3*t*x^2 + 1/(1+t^3)
%And plots the real function in addition to the approximation, and
%a graph of the error
    RHS = @(t,x) -3*t*x^2 + 1/(1+t^3)
    t = linspace(0,5,N+1);
    x = @(t) t/(1+t^3);
    y = arrayfun(x,t);
    plot(t,y)
    hold on
    adms = adams_pc5(RHS,0,0,5,N)
    diff = @(x,y) abs(x-y)
    err = arrayfun(diff,adms,y)
    plot(t, adms,'r--')
    figure()
    plot(t,err)
```



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

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

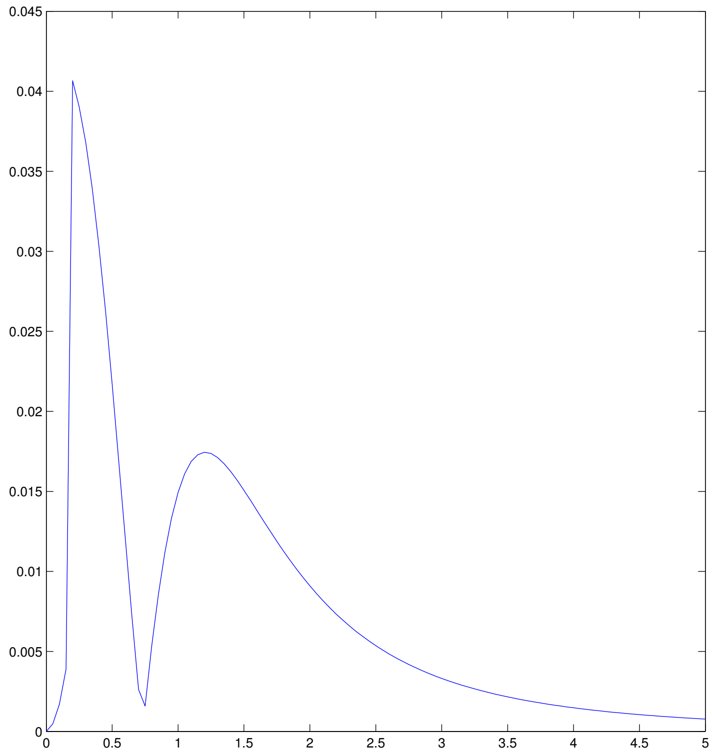
Sjá myndir 1 og 2.

1.3

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

1.4 Tímamælingar

```
numtests=100
adamstimes=[];
rkf45times=[];
rkv56times=[];

for i=1:numtests
    tic
    adams_pc5('Test',0,0,5,100);
    adamstimes(i)=toc;
```



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

```
    tic
    rkf45('Test',0,0,5,[0.0001,1,0.0001]);
    rkf45times(i)=toc;
    tic
    rkf45('Test',0,0,5,[0.00001,1,0.00001]);
    rkf45times(i)=toc;
    tic
    rkv56('Test',0,0,5,[0.00001,1,0.00001]);
    rkv56times(i)=toc;
end
adamsmeantime=sum(adamstimes)/numtests
rkf45meantime=sum(rkf45times)/numtests
rkv56meantime=sum(rkv56times)/numtests
```

Keyrsla á þessu forriti gaf eftirfarandi niðurstöður

fall	meðatími
adams_pc5	0,0082
rkf45	0,0024
rkv56	0,0027

2 Einfaldur pendúll

Við fáum að þar sem

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

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:

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

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

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

```
function y = pendull(RHS,lotur,n,theta0,theta1,omega,t0,res)
%% pendull.m
% Skipanaskra sem byr til hreyfimynd af einfoldum penduli, tekur
% inn ODE pendulsins sem RHS, hve margar lotur, hve margar myndir per lotu, upphafsgildin
% theta0,theta1, omega og t0, og hve margrfaltfleiri itranir af
% nalgun eru en rommum.
aviobj = avifile('pendull.avi','compression','None','fps',16); %#ok<REMFF1>
% Segir til um nafn myndbandsins, thjoppun og fjolda ramma a sek.
% Windows notendur aettu ad breyta 'None' i 'Indeo5' eda i einhvern annan
% compression moguleika.
% Mac, Linux, BSD og onnur styrikerfi thurfa ad thjappa myndbandid
% handvirkt utan matlab, t.d med ffmpeg.
fig=figure;
%% Fastar
%lotur = 5;          %Fjoldi lota til ad reikna
%n = 25;             %Fjoldi mynda i hverri lotu
%theta0 = 2;         %theta(t0)
%theta1 = 0;         %theta'(t0)
%omega = 1;          %Hornhradi
%t0 = 0;
%res = 6;
%% Jofnurnar
theta = @(t) theta0*cos(omega*(t-t0)) + (theta1/omega) * sin(omega*(t-t0));
dtheta = @(t) -omega*theta0*sin(omega*(t-t0))+ theta1*cos(omega*(t- ...
                                                t0));

%simple = adams_pc5('pendulODE',t0,[theta0,theta1],2*pi*lotur,res*lotur*n);
simple = adams_pc5(RHS,t0,[theta0,theta1],2*pi*lotur,res*lotur*n);
y = (max(simple(1,:))^2 + max(simple(2,:))^2)
thetasimple = @(t)  simple(1, res*floor(t*n/(2*pi)) + 1);
dthetasimple = @(t) simple(2, res*floor(t*n/(2*pi)) + 1);
%%
% Thar sem vid hreinsum myndina i hverju skrefi ta thurfum vid ad geyma
% fasahnitin i fylki
fasahnit = [theta(0);dtheta(0)];
simplefasahnit =[thetasimple(0);dthetasimple(0)];
for t = 0:2*pi/n:2*pi*lotur
    %% Fasaritid
    subplot(2,2,1) %Skipar matlab ad nota seinni hlutan af myndflotinum
```

```
    fasahnit = [fasahnit(1,:) theta(t); fasahnit(2,:) dtheta(t)]; %Baetir nyju ...
        fasahnitunum vid thau gomlu.
    simplefasahnit = [simplefasahnit(1,:) thetasimple(t);simplefasahnit(2,:) ...
        dthetasimple(t)]; %Baetir nyju fasahnitunum vid thau gomlu.
    plot(theta(t),dtheta(t),'ob', 'MarkerSize', 6) %Punkturinn i
                                                 %fasaritinu
    hold on % Thurfum ad setja "hold on" her svo vid yfirskrifum ekki linuna i fasaritnu ...
        bara med punktinum
    plot(fasahnit(1,:),fasahnit(2,:),'b') % Linan i fasaritinu
    hold on
    plot(thetasimple(t),dthetasimple(t),'or', 'MarkerSize', 6)
    axis([-theta0-0.2,theta0+0.2,-theta0-0.2,theta0+0.2])
    axis square
    hold off
    subplot(2,2,2) %Skipar matlab ad nota seinni hlutan af myndflotinum
    plot(simplefasahnit(1,:),simplefasahnit(2,:),'r') % Linan i fasaritinu
    hold on
    plot(thetasimple(t),dthetasimple(t),'or', 'MarkerSize', 6)
    %Punkturinn i fasaritinu
    hold on
    plot(theta(t),dtheta(t),'ob', 'MarkerSize', 6) %Punkturinn i
    axis([-theta0-0.2,theta0+0.2,-theta0-0.2,theta0+0.2])
    axis square
    hold off
    %% Pendullinn
    %Skiptir myndaflotinum i 2x1 fylki og segir matlab ad nota fyrsta stakid
    % Teikniskipun fyrir pendulinn: teiknum linu fra [0,0] (festipunktur)
    % i [sin(theta(t)),-cos(theta(t))] sem er stadsetning lodsins
    % '-o' segir ad vid aetlum ad teikna linu med hringlaga endapunkta
    % 'MarkerSize' setur staerd endapunktanna
    % 'MarkerFaceColor' akvardar lit endapunktana
    subplot(2,2,3)
    plot([0,sin(theta(t))],[0,-cos(theta(t))],'-o','MarkerSize',8, ...
         'MarkerFaceColor','b')
    axis([-1.2,1.2,-1.2,1.2]) %Festir asana
    axis square %Thvingar matlab til ad hafa x og y asinn jafn
                %langan
    hold off
    subplot(2,2,4)
    plot([0,sin(thetasimple(t))],[0,-cos(thetasimple(t))],'-o','MarkerSize',8,'MarkerFaceColor','r')
    axis([-1.2,1.2,-1.2,1.2]) %Festir asana
    axis square %Thvingar matlab til ad hafa x og y asinn jafn
                %langan
    hold off
    %% Hreyfimynd
    F = getframe(fig); %Naer i nyjasta ramman
    aviobj = addframe(aviobj,F); % Og skeytir thvi vid restina
end
%% Fragangur
close(fig); %Lokar myndinni til ad ekki se haegt ad yfirskrifa hana
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:

```
%%hr2
%fall sem keyrir pendul og byr til hreyfimynd med miklu utslagi
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 útslagið, 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 útslagið. Munurinn liggur í mismunandi fasahliðrunum.

```
function y = roluODEmin (t,x)
    %j fnuhneppi sem l sir r lu sem h gir hva  mest    s r
    g = 1;
    l = 1;
    utslag = 0.8;
    horntidni = sqrt(g/l)/(2*pi*(1+ (x(2)^2)/16 + (x(2)^4)*11/3072));
    fasahlidrun = 0;
    s = @(t) l + utslag*cos(horntidni * t + fasahlidrun);

    y(1) = x(2);
    y(2) = -(g/s(t))*sin(x(1));

end
```

```
function y = roluODEmax (t,x)
    % j fnuhneppi sem l sir r lu sem eykur sveifluna sem mest
    g = 1;
    l = 1;
    utslag = 0.8;
    horntidni = sqrt(g/l)/(2*pi*(1+ (x(2)^2)/16 + (x(2)^4)*11/3072));
    fasahlidrun = pi;
    s = @(t) l + utslag*cos(horntidni * t + fasahlidrun);

    y(1) = x(2);
    y(2) = -(g/s(t))*sin(x(1));

end
```

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

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

.

Útslagið örvaðist mest þegar $\omega = 1/T$ eins og gefið var í 3 örvunin var jákvæð (sveiflan jókst) þegar fasahornið var π í byrjun en neikvæð ef fasahornið var $\varphi = 0$, þá hægði maður á rólunni. Munurinn liggur í hvenær maður styttir og lengir bandið, en ef fasahornið er π, þá er mesta lengingin í miðjunni og minnst í endapunktunum, en öfugt er farið þegar hægt er á rólunni.

Orsökina fyrir þessari hegðun í okkar einfalda módeli er aðfinna í orkubúskap kerfisins en við gerum ráð fyrir að enginn orka fari í aðhreyfa massamiðjuna upp og niður róluna

þegar rólan er í hæstu stöðu þá er öll orka kerfisins stöðuorka og því ofar sem massamiðjan er þeim mun meiri er orka kerfisins og á svo lengi sem sveiflan er ekki meiri en $\frac{\pi}{2}$ frá lóðréttu þá færir það massamiðjuna ofar að stytta róluna

eins þegar rólan er lárétt (í mijunni) þá er öll orka kerfisins skriðorka og því neðar sem massamiðjan er því meiri stöðuorka hefur veriðlosuð í skriðorku og augljóslega er massamiðjan neðar þegar lengt er í rólunni í

þessari stöðu
þegar fasahorninu er breytt þannig að rólan sé styðst í lóðréttu stöðunni og lengst í mesta útslaginu þá er ekki öll orkan í kerfinu færð milli skriðorku og stöðuorku í hverri sveiflu og því glatast orka úr kerfinu
Rétt er að athuga að þetta brýtur ekki gegn orkuvarðveislulögmálinu af því að sú orka sem hverfur (eða verður til þegar veriðvar að auka sveifluna) fer í vinnuna að færa massamiðjuna upp og niður róluna
einnig sést í forritinu fyrir jákvæða örvun að þegar útslagið er orðið meira en $\frac{\pi}{2}$ þá er næsta sveifla á eftir ekki hærri þetta er vegna þess að þegar sveiflan er orðin þetta stór þá færist massamiðjan neðar viðþaðað stytt sé í rólunni og því er minni stöðuorka til staðar til þess að breyta í skriðorku fyrir næstu sveiflu. (þessar upplýsingar hefðu veriðvel þegnar fyrir um 15 árum síðan þegar undirritaður var aðreyna að ná heilum hring í rólunum á leikvellinum)

4 Kúlupendúll

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

```
function y = kulupendulODE(t,x)
    m = 1;
    l = 1;
    g = 9.82;
    om = 1/(m*l*l);

    y(1) = om*x(3);
    y(2) = om*x(4)/(sin(x(1))^2);
    y(3) = om*x(4)^2*(cos(x(1))/(sin(x(1))^3)) - m*l*g*sin(x(1));
    y(4) = 0;
```

Eftirfarandi forritsbútur var notaður til að búa til myndbandið.

```
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.

```
function y = pendull3d(RHS,lotur,n,x0,t0,res)
%% pendull.m
% Skipanaskra sem byr til hreyfimynd af einfoldum penduli, tekur
% inn ODE pendulsins sem RHS, hve margar lotur, hve margar myndir per lotu, upphafsgildin
% theta0,theta1, omega og t0, og hve margrfaltfleiri itranir af
% nalgun eru en rommum.
aviobj = avifile('pendull.avi','compression','None','fps',16); %#ok<REMFF1>
% Segir til um nafn myndbandsins, thjoppun og fjolda ramma a sek.
% Windows notendur aettu ad breyta 'None' i 'Indeo5' eda i einhvern annan
% compression moguleika.
% Mac, Linux, BSD og onnur styrikerfi thurfa ad thjappa myndbandid
% handvirkt utan matlab, t.d med ffmpeg.
fig=figure;
%% Fastar
%lotur = 5;          %Fjoldi lota til ad reikna
%n = 25;             %Fjoldi mynda i hverri lotu
theta0 = 2;         %theta(t0)
theta1 = 0;         %theta'(t0)
%omega = 1;          %Hornhradi
%t0 = 0;
%res = 6;
%% Jofnurnar

simple = adams_pc5(RHS,t0,x0,2*pi*lotur,res*lotur*n);
theta = @(t)  simple(1, res*floor(t*n/(2*pi)) + 1);
phi = @(t)  simple(2, res*floor(t*n/(2*pi)) + 1);
ptheta = @(t) simple(3, res*floor(t*n/(2*pi)) + 1);
pphi = @(t) simple(4, res*floor(t*n/(2*pi)) + 1);
```

```
%%
% Thar sem vid hreinsum myndina i hverju skrefi ta thurfum vid ad geyma
% fasahnitin i fylki
%fasahnit = [theta(0);dtheta(0)];
fasahnit = [theta(0);phi(0)];
x = @(t) sin(theta(t))*cos(phi(t));
y = @(t) sin(theta(t))*sin(phi(t));
z = @(t) -cos(theta(t)); %theta er midad vid ofugan z-as en venjulega
for t = 0:2*pi/n:2*pi*lotur
    %% Fasaritid
    %subplot(2,1,1) %Skipar matlab ad nota seinni hlutan af myndflotinum

    fasahnit = [fasahnit(1,:) x(t); fasahnit(2,:) y(t)]; %Baetir nyju fasahnitunum vid ...
        thau gomlu.

    %plot(x(t),y(t),'ob', 'MarkerSize', 6) %Punkturinn i
                                                %fasaritinu

    %hold on % Thurfum ad setja "hold on" her svo vid yfirskrifum ekki linuna i fasaritnu ...
        bara med punktinum

    %plot(fasahnit(1,:),fasahnit(2,:),'b') % Linan i fasaritinu
    %hold on
    %axis([-1.2,1.2,-1.2,1.2])
    %axis square

    %hold off
    %% Pendullinn
    %Skiptir myndaflotinum i 2x1 fylki og segir matlab ad nota fyrsta stakid
    % Teikniskipun fyrir pendulinn: teiknum linu fra [0,0] (festipunktur)
    % i [sin(theta(t)),-cos(theta(t))] sem er stadsetning lodsins
    % '-o' segir ad vid aetlum ad teikna linu med hringlaga endapunkta
    % 'MarkerSize' setur staerd endapunktanna
    % 'MarkerFaceColor' akvardar lit endapunktana


    %subplot(2,1,2)
    plot3([0,x(t)],[0,y(t)],[0,z(t)],'-o','MarkerSize',8,'MarkerFaceColor','b')
    hold on
    plot3(fasahnit(1,:),fasahnit(2,:),zeros(1,length(fasahnit(2,:)))-1,'r', ...
          'MarkerSize',8,'MarkerFaceColor','r')
   % hold off
    grid %minor
    axis([-1.2,1.2,-1.2,1.2,-1.2,1.2]) %Festir asana
    axis square %Thvingar matlab til ad hafa x og y asinn jafn
    hold off
    %% Hreyfimynd
    F = getframe(fig); %Naer i nyjasta ramman
    aviobj = addframe(aviobj,F); % Og skeytir thvi vid restina
end
%% Fragangur
close(fig); %Lokar myndinni til ad ekki se haegt ad yfirskrifa hana
aviobj = close(aviobj); %Lokar og byr til myndbandid
```

5 Lorenz Attractor

Við ákváðum að gera myndaband af Lorentz Attractorum, en það eru lausnir á diffurjöfnum sem gefa ákaflega fallega ferla. Diffurjöfnuna má finna hér, en við henni var bættur möguleiki til að taka inn fasta sem notaðir eru, en það er til þess að auðvelda okkur að teikna margar eindir í einu.

```
function y = lorenzODE (t,x,cond)
    %fall sem er j fnuhneppi lorenz kerfis m e
    %t er breytan
    %x er vigur m e  upphafs og endaskilyr um
```

```
    %cond er vigur sem inniheldur stika lorenz kerfisins
    sigma = cond(1);
    r = cond(2);
    b = cond(3);
    y(1) = sigma* (x(2) - x(1));
    y(2) = x(1) * (r - x(3)) - x(2);
    y(3) = x(1)* x(2) - b*(x(3));
```

Eftirfarandi forrit var svo búið til til þess að gera myndbandið, en það tekur sem viðföng fastana og upphafsstaðsetningar eindanna, svo auðvelt væri að bæta við fleirum.

```
function y = lorenzAnim(lotur,n,res,cond,points,markercolors,plotcolors)
%% lorenzAnim.m
% Skipanaskra sem byr til hreyfimynd af lorenz attractor,
% tekur inn hve lengi a ad keyra, hve margir rammar a sekundu,
% hversu margfalt meira er renderad af gildum, upphafsgildi
% einda a formi lista [[sigma1;r1;b1], [sigma2;r2;b2]] etc
% upphafs stadsetningar  eirra  a formi
% [[x0,y0,z0];[x0_2,y0_2,z0_2];...] etc sama lengd og
% upphafsskilirdin
% og svo tvo vigra markercolors og plotcolors sem segja til um
% litina a eindunm og ferlunum sem thaer skilja eftir sig.

aviobj = avifile('lorenz.avi','compression','None','fps',16); %#ok<REMFF1>
% Segir til um nafn myndbandsins, thjoppun og fjolda ramma a sek.
% Windows notendur aettu ad breyta 'None' i 'Indeo5' eda i einhvern annan
% compression moguleika.
% Mac, Linux, BSD og onnur styrikerfi thurfa ad thjappa myndbandid
% handvirkt utan matlab, t.d med ffmpeg.
fig=figure;

s = size(cond);
fjoldi = s(2);
%Byr til follin og reiknar ut nalgunina a theim
for i = 1:fjoldi
    lorenzFunc = @(t,x) lorenzODE(t,x,cond(:,i));
    Sol(i, : , :) = adams_pc5(lorenzFunc,0,points(i,:),lotur,res*lotur*n);
end

for t = 0:1/n:lotur

    curr = res*floor(t*n)+1;
    % Teiknar inn eindirnar
    for i = 1:fjoldi
        Solv = Sol(i,:,1:curr);
        plot3(Solv(1,1,curr),Solv(1,2,curr),Solv(1,3,curr), markercolors(i), 'MarkerSize', 6)
        hold on
        x = Solv(1,1,:);
        y = Solv(1,2,:);
        z = Solv(1,3,:);
        plot3(x(:),y(:),z(:),plotcolors(i))
        hold on
    end
    view(30+2*t,30+2*t)
    grid %minor
    hold off
    %% Hreyfimynd
    F = getframe(fig); %Naer i nyjasta ramman
    aviobj = addframe(aviobj,F); % Og skeytir thvi vid restina
end
%% Fragangur
close(fig); %Lokar myndinni til ad ekki se haegt ad yfirskrifa hana
aviobj = close(aviobj); %Lokar og byr til myndbandid
```

Eftirfarandi forritsbútur var svo gerður sem býr til myndbandið sem skilað var inn.

```
%Forritsbutur sem byr til myndbandid af Lorentz attractor kerfi
%Gaeti thurft ad minnka resid, en thetta gefur allavegana frekar smooth feril svona
lorenzAnim(30,32,2^11,[[10;28;8/3],[10;65;12],[10;14;8/3],[10;0.5;8/3]], ...
    [[5,5,5];[5,5,5];[5,5,5];[5,5,5]], ['or', 'og', 'ob','oy'], ['r','g','b','y']);
```

Að skýrsluni unnu : ____________________ ____________________ ____________________