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

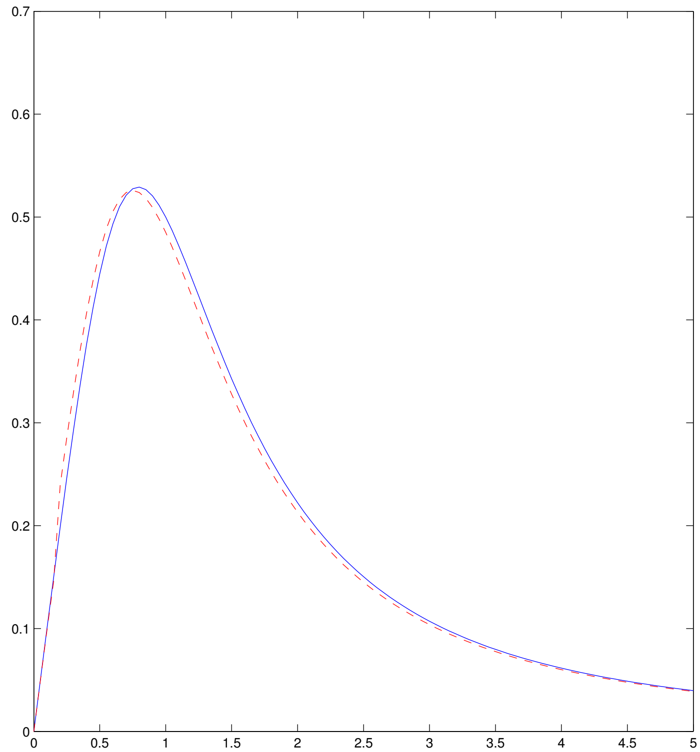
### 1.1

```
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';
```

```
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

```
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 rauttbrotastrik

```
%%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)
```

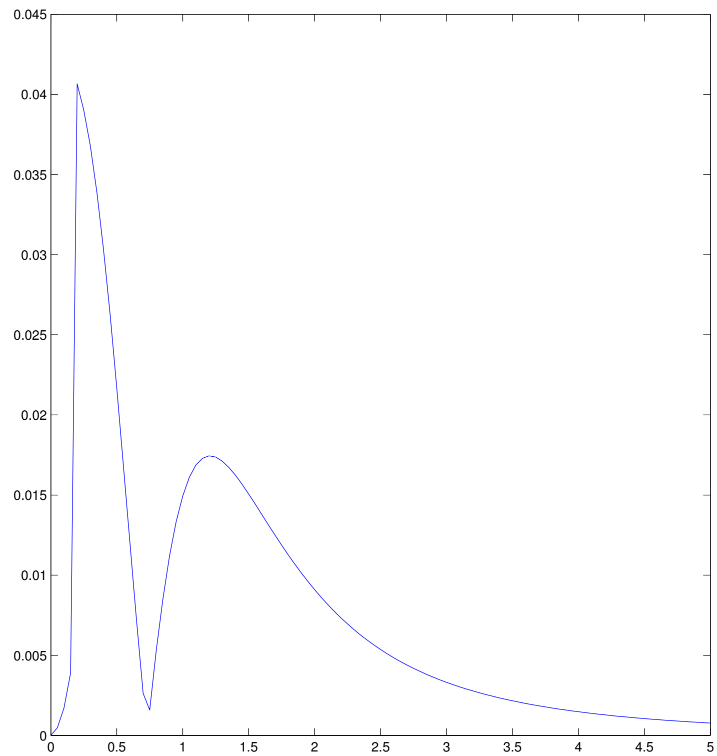
Sjá myndir 1 og 2.

### 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

## 4 Kúlupendúll

## 5 Sólkerfi

Að skýrsluni unnu : ____________________ ____________________ ____________________