Bjarki Geir Benediktsson,
Haukur Óskar Þorgeirsson,
Matthías Páll Gissurarson

Töluleg Greining

Heimaverkefni 2

8. mars 2013

Töluleg Greining
Heimaverkefni 2

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

8. mars 2013

Inngangur

Verkefni þetta var í tveimur pörtum. Í þeim fyrri vinnum við með tvær brúunaraðferrðir, splæsibrúun og Bezier brúun. Við útfærum splæsibrúun með hinum ýmsu endaskilyrðum og prufum svo að teikna nokkra punkta og sjá hvað gerist.

Svo útfærum við Bezier brúun sem hægt er að nota sem teikniforrit.

Í öðrum hluta útfærum við almenna aðferð við útgiskun og beitum henni til að útfæra Richardson, Romberg og nálganir á stigli og Hesse-fylki.

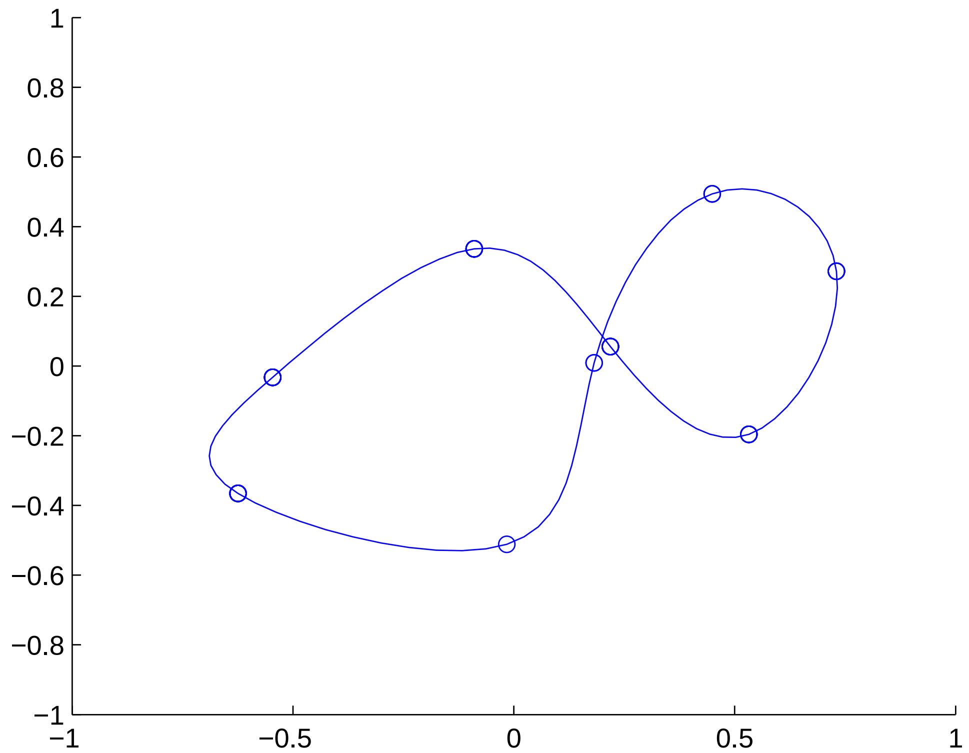
1 Ferilsteikning með splæsibrúun

1.1 Teikning með aflestri af skjá

```
% Forrit sem utfaerir splaesibruun med lotubundnum endaskilyrdum
    close all
    clear all

    % Setjum mynd upp
    a=-1; b=1; c=-1; d=1;
    axis([a b c d])

    % Lesum inn punkta
    hold on
    hnappur=1;
    x = []; y = [];

    while hnappur==1
        [xtmp,ytmp,hnappur]=ginput(1);
        if hnappur==1
            x = [x, xtmp];
            y = [y, ytmp];
            plot(x,y,'o')
        end
    end
    x=[x, x(1)];
    y=[y, y(1)];
    % Stikum ferilinn svo vid lendum ekki i veseni
    % ef x-hnitin eru ekki i staerdarrod
    n = length(x);  t = 1:n;  tt=linspace(1,n,100);

    % Reiknum og teiknum
    xx = splaesi(t,x,4,0,0,tt);
    yy = splaesi(t,y,4,0,0,tt);
```

```

    plot(xx,yy)
```

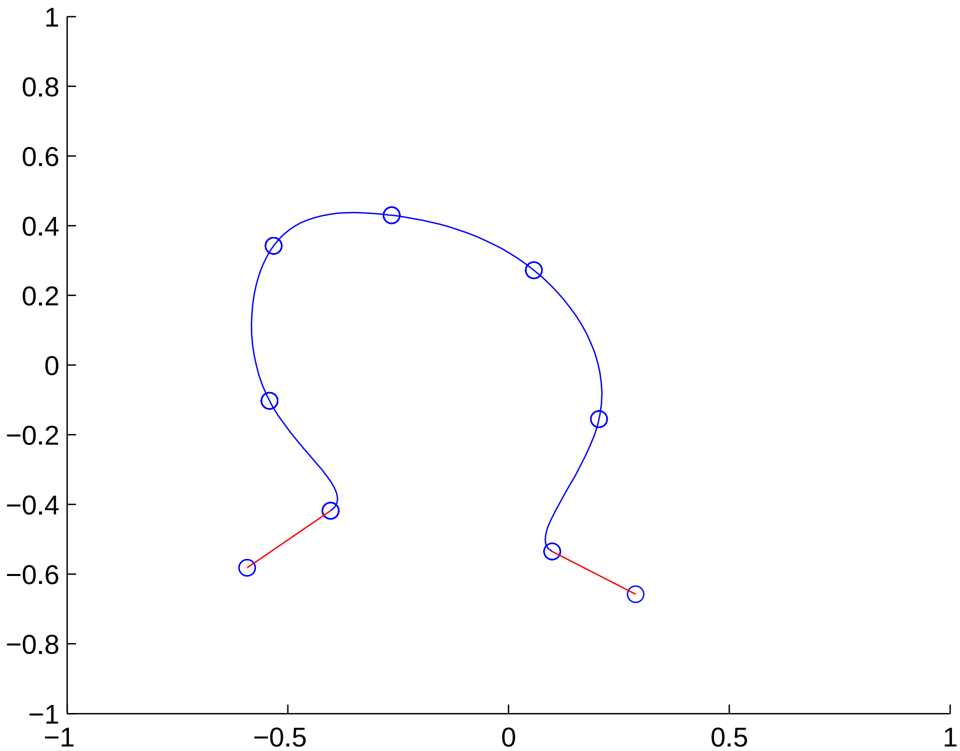
1.2 Teikning á lokuðum ferlum með aflestri af skjá



1.2.1 Teikning á þvinguðum ferlum með aflestri af skjá

```
%Forrit sem utfaerir splaesibruun med thvingudum endaskilyrdum
    close all
    clear all

    % Setjum mynd upp
    a=-1; b=1; c=-1; d=1;
    axis([a b c d])

    % Lesum inn punkta
    hold on
    hnappur=1;
    x = []; y = [];

    while hnappur==1
        [xtmp,ytmp,hnappur]=ginput(1);
        if hnappur==1
            x = [x, xtmp];
            y = [y, ytmp];
            plot(x,y,'o')
        end
    end
```

```
    df1x = (x(1)-x(end-1));
    df1y = (y(1)-y(end-1));
    df2x = (x(end) - x(end-2));
    df2y = (y(end) -y(end-2));
    plot([x(1) x(end-1)], [y(1) y(end-1)], 'r')
    plot([x(end-2) x(end)], [y(end-2) y(end)], 'r')
    x = x(1:end-2);
    y = y(1:end-2);

    % Stikum ferilinn svo vid lendum ekki i veseni
    % ef x-hnitin eru ekki i staerdarrod
    n = length(x);  t = 1:n;  tt=linspace(1,n,100);

    % Reiknum og teiknum
    %Tharf ad reikna ut c0 og cn
    xx = splaesi(t,x,2,df1x,df2x,tt);
    yy = splaesi(t,y,2,df1y,df2y,tt);

    plot(xx,yy)
```



1.3 Ferilteikning með Bezier-splæsibrúun

```
% Teikniforrit sem notar bezier-bruun milli punkta sem teiknadir eru til ad
% teikna myndir.
    close all
    clear all

    % Setjum mynd upp
```

```
    a=-1; b=1; c=-1; d=1;
    axis([a b c d])

    % Lesum inn punkta
    hold on
    hnappur=1;
    x = []; y = [];
    n=4;
    t=0:1;
    tt=linspace(0,1,100);
    xxx = []; %Hehe
    yyy = [];

    while hnappur==1
        [xtmp,ytmp,hnappur]=ginput(1);
        if hnappur==1
            x = [x, xtmp];
            y = [y, ytmp];
            plot(x,y,'o')
        end
        if hnappur==1&length(x) == 4
            df1x = (x(2)-x(1));
            df1y = (y(2)-y(1));
            df2x = (x(4) - x(3));
            df2y = (y(4) - y(3));
            plot([x(2) x(1)], [y(2) y(1)], 'r')
            plot([x(4) x(3)], [y(4) y(3)], 'r')
            xx = baz(x,tt);
            xxx = [xxx xx];
            yy = baz(y,tt);
            yyy = [yyy yy];
            plot(xx,yy)

            x = [x(4) (2*x(4)-x(3))];
            y = [y(4) (2*y(4)-y(3))];
        end
    end


    close all
    a=-1; b=1; c=-1; d=1;
    axis([a b c d])

    % Lesum inn punkta
    hold on
    plot(xxx,yyy)
```

Fallið baz þjónar sama tilgangi fyrir Bezier brúunina og splaesi gerði fyrir splæsibrúunina það tekur við lista af fjórum hnitum $(x_i)_{i=0}^3$ og lista $(t_i)_{i=1}^n$ af gildum á $[0,1]$ og skilar lista af gildum $(r(t_i))_{i=1}^n$ þar sem r

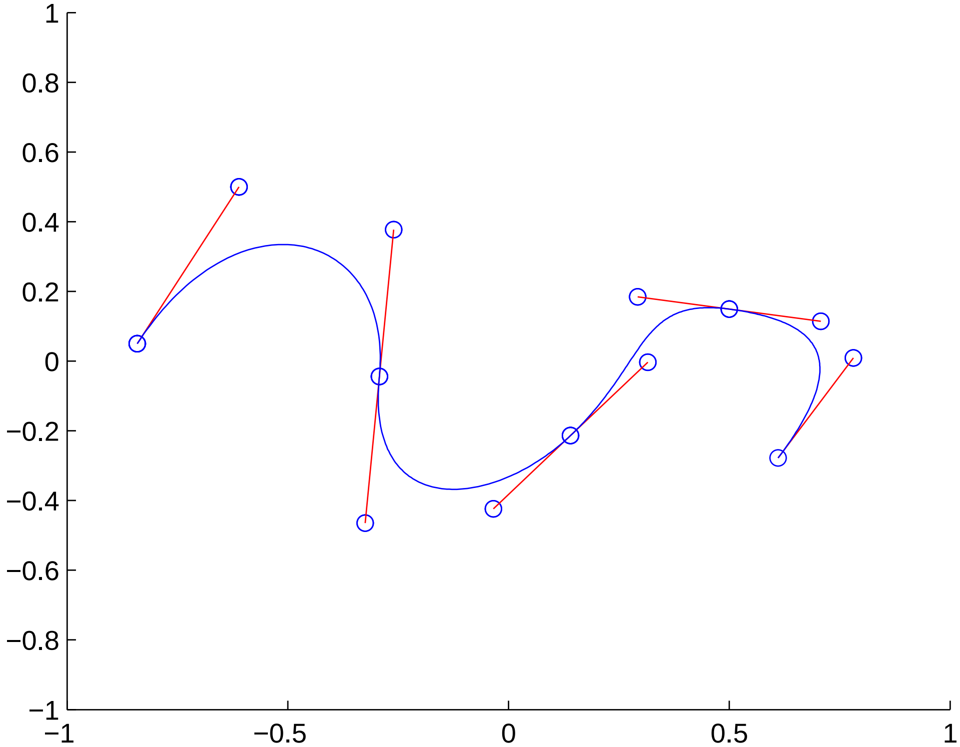
$$r(t) = (1-t)^3 x_0 + 3(1-t)^2 t x_1 + 3(1-t)t^2 x_2 + t^3 x_3$$

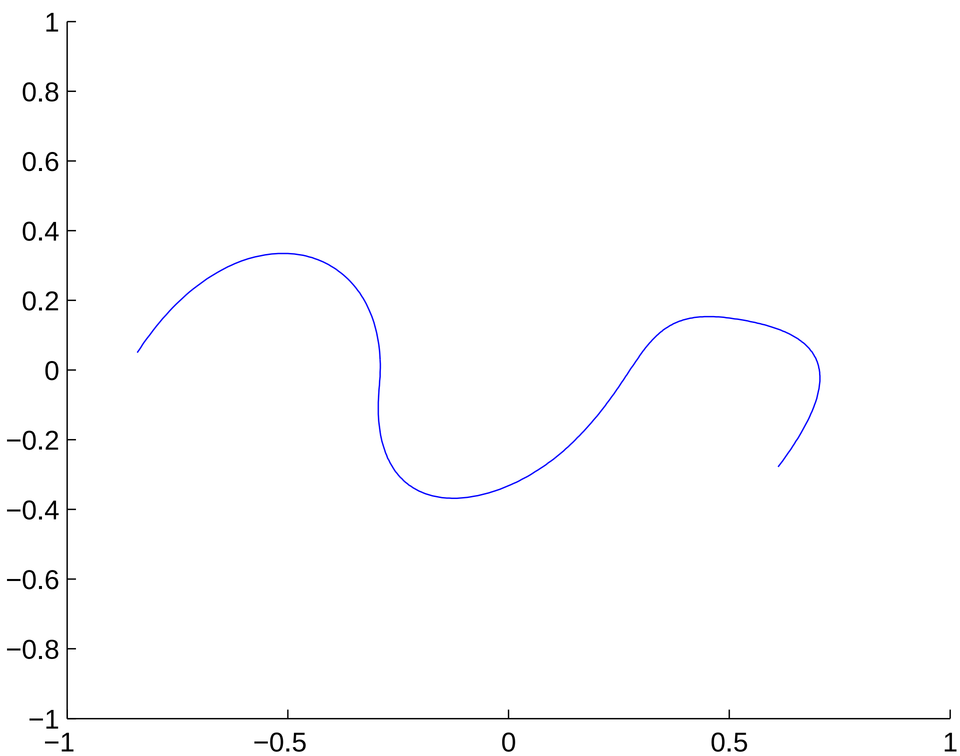
nú fæst að með því að gefa baz y-hnit í stað x-hnita sem inntak skilar það lista af $s(t)$ gildum þar sem

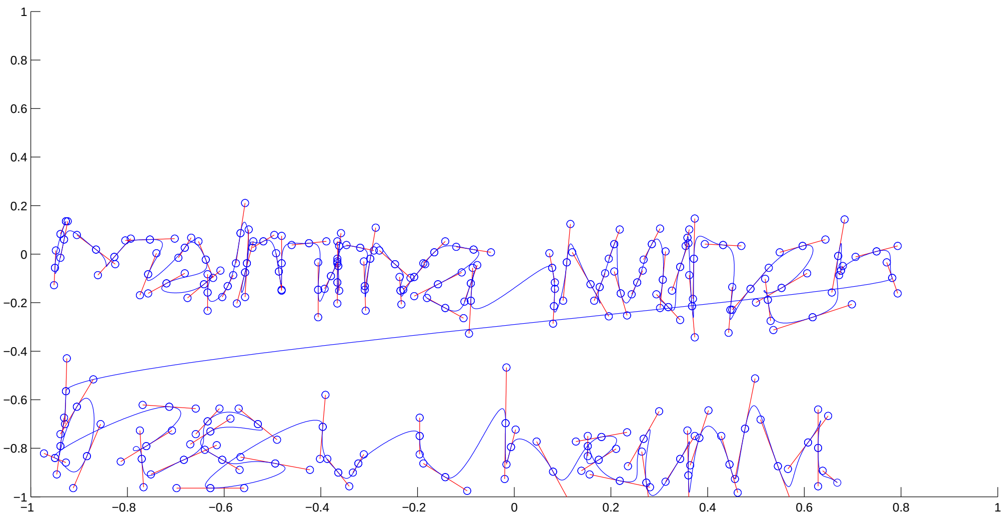
$$s(t) = (1-t)^3 y_0 + 3(1-t)^2 t y_1 + 3(1-t)t^2 y_2 + t^3 y_3$$

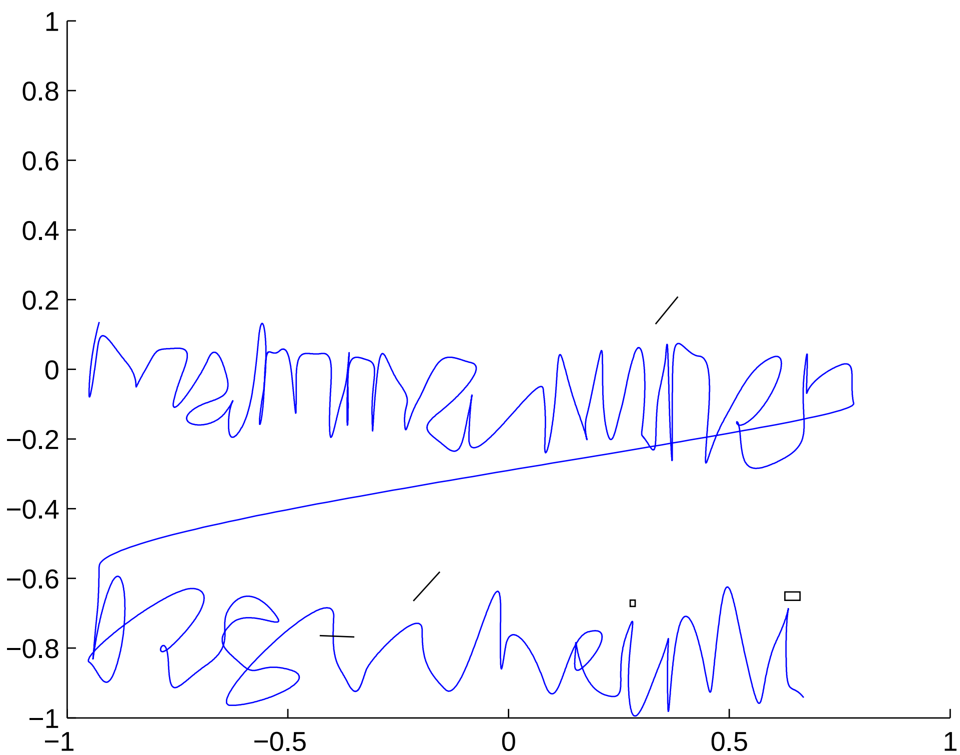
en það eru þau gildi sem við þurfum til að geta framkvæmt Bezier brúunina

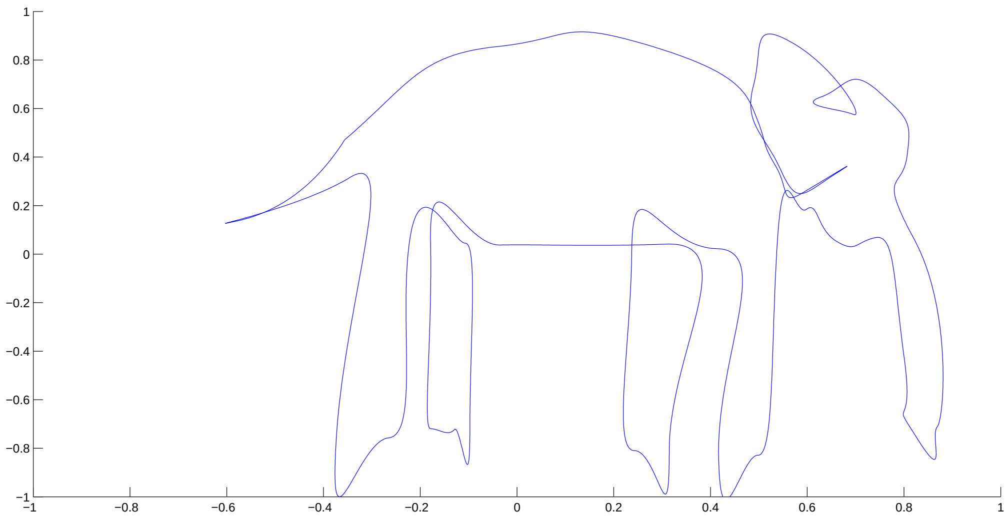
```
%Fall sem reiknar ut gildin i ferlinum fyrir bezier bruun

function yy = baz( y, tt)
    for i=1:length(tt)
        t=tt(i);
        yy(i)=(1-t)^3*y(1)+3*(1-t)^2*t*y(2)+3*(1-t)*t^2*y(3)+t^3*y(4);
    end
end
```


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−1
−0.5
0
0.5
1


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−1
−0.5
0
0.5
1







hér mætti leyfa notenda að rjúfa ferilin með það er að vera með punkta sem ekki væri teiknað á milli Auk þess væri mjög sniðugt að geta hreyft punktana eftir að þeir hafa verið settir inn til að laga ferilin til

2 Nálgun á afleiðum,heildu, stiglum og Hessefylkjum

2.1 Almenn útgiskun

```
%
% Matlab-forrit sem reiknar ut nalgun a hverju thvi sem R.m skilgreinir
% med utgiskun
%
%
% Inn fara: fallid f, sem skilgreint er med fallbreytu
%           a - punkturinn,
%           h - upphafsgildi a skreflengd,
%           imax - hamarksfjoldi itrekana,
%           epsilon - nakvaemniskrafan.
% Ut koma:  X - aftasta gildid i Richardson utgiskunartoflunni.
%           mat1 - svartsyna eftiramatid a skekkju, D(i,i)-D(i-1,i-1).
%           mat2 - brjartsyna eftiramatid a skekkju, D(i,i)-D(i,i-1).
%
function [X,mat1,mat2]=extrapolation(f,a,h0,imax,epsilon)
h=h0;
D=cell(imax,imax);
D{1,1}=R(f,a,h);
i=2;
h=h/2;
mat1=2*epsilon;
while i≤imax & abs(mat1)>epsilon
    D{i,1}=R(f,a,h);
    for j=2:i
        mat2=(1/(4^(j-1)-1))*(D{i,j-1}-D{i-1,j-1});
        D{i,j}=D{i,j-1}+mat2;
    end
    mat1=norm(D{i,i}-D{i-1,i-1});
    X=D{i,i};
    i=i+1;
    h=h/2;
end
mat2=norm(mat2);
```

```
end
```

2.2 Richardson

Prófun fyrir f'(0): R fallið:

```
function r = R(f,a,h)
    r = (f(a+h) + f(a−h))/(2*h);
end
```

Úr extrapolation kom:

```
>>[X,mat1,mat2]=extrapolation(@(x)sin(x), 0, 0.5, 100, 1e−10)
X = 1
mat1 = 2.6201e−012
mat2 = 1.0235e−014
```

og úr Richardson kom:

```
>> [X,mat1,mat2]=richardson(@(x)sin(x), 0, 0.5, 100, 1e−10)

X = 1

mat1 = 2.6201e−012

mat2 = 1.0235e−014
```

Prófun fyrir f"(0): R fallið:

```
function r = R(f,a,h)
  r = (f(a+h) + f(a−h) − 2*f(a))/(h*h);
end
```

úr extrapolate kom

```
>> [X,mat1,mat2]=extrapolation(@(x)sin(x), 0, 0.5, 100, 1e−10)
X = 0
mat1 = 0
mat2 = 0
```

Þar sem Richardson reiknar bara f', þannig að skiptum út fyrir cos(x) = sin'(x) til þess að prófa, en það gaf:

```
>> [X,mat1,mat2]=richardson(@(x)cos(x), 0, 0.5, 100, 1e−10)
X = 0
mat1 = 0
mat2 = 0
```

2.3 Romberg

R fallið:

```
function r = R(f,a,h,n)
    r=0;
    for i=0:n
        r = r + h*f(a+i*h);
```

```
    end
    r = r - h/2*(f(a) + f(a+n*h));
end
```

Útkoma Keyrsla með $\sin(x)$

```
>> [X,mat1,mat2]=extrapolation(@(x) sin(x),0,pi/2,100,1e-10)
X = 1.0000
mat1 = 1.9832e-012
mat2 = 1.9368e-015
```

Annað dæmi með e^x

```
>> [X,mat1,mat2]=extrapolation(@(x) exp(x),0,1,100,1e-10)
X = 1.7183
mat1 = 3.2863e-014
mat2 = 3.2124e-017
```

2.4 Nálgun á stiglum

```
function r = R(f,a,h)
    r=[(f(a+h*[1 0]) - f(a-h*[1 0]))/(2*h);(f(a+h*[0 1]) - f(a-h*[0 1]))/(2*h)];
end
```

Keyrsla með $f(x) = x_1^2 - x_2^2$ og punktinn (1,1)

```
>> [X,mat1,mat2]=extrapolation(@(x) x(1)^2 - x(2)^2,[1 1],1,100,1e-10)
X(:,:,1) = 2
X(:,:,2) = -2
mat1 = 0
mat2 = 0
```

Annað dæmi með $sin(x_1) * cos(x_2)$:

```
>> [X,mat1,mat2]=extrapolation(@(x) sin(x(1))*cos(x(2)),[1 1],1,100,1e-10)
X(:,:,1) = 0.2919
X(:,:,2) = -0.7081
mat1 = 1.7925e-014
mat2 = 1.7522e-017
```

2.5 Nálgun á Hessefylkjum

R fallið:

```
function r = R(f,a,h)
    A = (f(a + h*[1 0]) +f(a-h*[1 0]) - 2*f(a))/(h*h);
    C = (f(a + h*[0 1]) +f(a-h*[0 1]) - 2*f(a))/(h*h);
    B = (f(a + h*[1 0] + h*[0 1]) - f(a + h*[1 0] - h*[0 1]) - f(a - h*[1 0] + h*[0 1]) + ...
        f(a - h*[1 0] - h*[0 1]))/(4*h*h);
    r=[A B; B C];
end
```

Keyrsla með $f(x) = x_1^2 - x_2^2$ og punktinn (1,1)

```
>> [X,mat1,mat2]=extrapolation(@(x) x(1)^2 - x(2)^2,[1 1],1,100,1e-10)
X =  2     0
```

```
      0    −2
mat1 = 0
mat2 = 0
```

Annað dæmi með $sin(x_1) * cos(x_2)$:

```
>> [X,mat1,mat2]=extrapolation(@(x) sin(x(1))*cos(x(2)),[1 1],1,100,1e−10)
X =−0.4546   −0.4546
   −0.4546   −0.4546
mat1 = 1.8373e−012
mat2 = 1.7942e−015
```

Að skýrsluni unnu : ______________________ ______________________ ______________________