Bjarki Geir Benediktsson,
Haukur Óskar Þorgeirsson,
Matthías Páll Gissurarson

Töluleg Greining

Heimaverkefni 1
8. febrúar 2013

Töluleg Greining
Heimaverkefni 1

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

8. febrúar 2013

Inngangur

Verkefni þetta snýst um að nota matlab til þess að leita að stöðupunktum í gefnu falli og flokka þá. fyrst með því að leita handvirkt með því að teikna kassa utan um mögulega stöðupunkta út frá jafnhæðarferlum fallsins, og hins vegar með því að leita skipulega fyrir innan gefinn ramma.

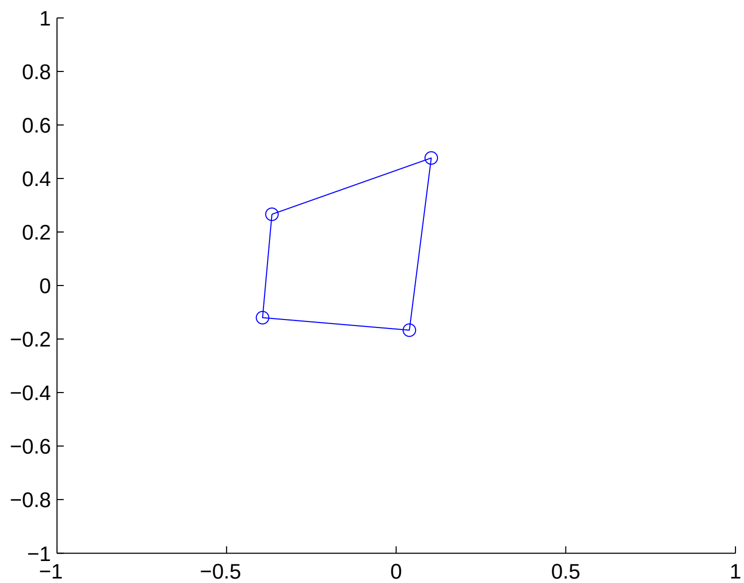
1 Innlestur hnita frá mús

Hér má sjá fyrsta forritið, en það má til dæmis keyra með `square(−1,1,−1,1)` Til að keyra það á $[-1, 1] \times [-1, 1]$. Þetta forrit virkar þannig að það kemur upp mynd af hnitakerfi þar sem maður getur valið fjóra punkta með þvi að smella á hnitakerfið. Svo er teiknaður ferhyrningurinn sem punktarnir skilgreina (gefið að hann sé kúptur, annars kemur upp villa), og innan forritsins eru punktarnir komnir í þannig röð að þeir ganga réttsælis í ferhyrningnum. Forritið hættir ef smellt er á hægri músarhnapp.

```
%Tekur inn hnit med musasmellum og skilar fylki med theim i rettri rod.
function P = square(a,b,c,d)
    %Teiknum hnitakerfi
    axis([a b c d])
    hold on
    hnappur = 1;
    i = 1;
    %Buum til fylki til a   geyma ni urst  ur.
    %P = [ 0 0 0 0 0; 0 0 0 0 0];
    %Fylki er buid til sjalfkrafa
    while hnappur == 1 && i ≤ 4;
        [x,y,hnappur] = ginput(1);
        %Viljum bara f   4 punkta.
        if hnappur == 1;
            P(1,i) = x;
            P(2,i) = y;
            plot(x,y,'o');
            i = i+1;
        end
    end
    if i ≤ 4
        %Getum notad thetta til ad akvarda hvort haegri klikk kom.
        P = [];
        return
    end
    %Possum ad vid hofum fengid nogu marga punkta
    if length(unique(P.','rows')) ≥ 4
        P = P(:,flipdim(convhull(P(1,1:4)',P(2,1:4)'),1));
        plot(P(1,:),P(2,:))
        %Possum ad vid seum med kassa en ekki t.d. Thrihyrning
        if length(P) < 5
```

```
            P = square(a,b,c,d);
        end
    else
        %Ef ekki bidjum vid um nyjan kassa
        P = square(a,b,c,d);
    end
end
```

Keyrt með `square(-1,1,-1,1)` fæst:



Mynd 1: Ferhyrningur teiknaður

```
ans =

  Columns 1 through 4

   -0.3664    0.1037    0.0392   -0.3940
    0.2661    0.4766   -0.1667   -0.1199

  Column 5

   -0.3664
    0.2661
```

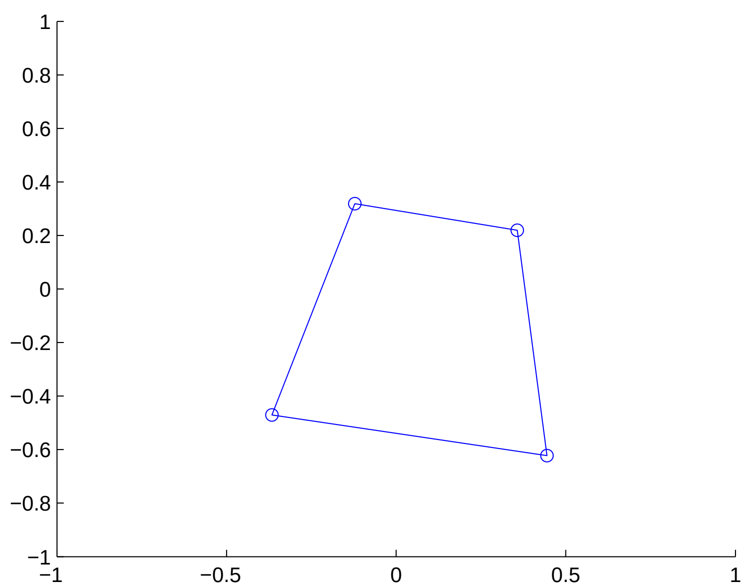
2 Er þessi punktur inni í þessum kassa?

Hér er forrit sem vinnur meira á bak við tjöldin. Það reiknar, fyrir gefinn punkt (dálkvigur) og gefinn ferhyrning (gefinn með hornpunktunum sem eru raðaðir réttsælis) hvort punkturinn sé inni í ferhyrningnum.

```
function flag = square_check(x,P)

v = zeros(2,4);
for i = 1:4;
    v(1:2,i) = P(1:2,i+1) - P(1:2,i);
```

```
end

n = zeros(2,4);

for i = 1:4
    n(1,i)=-v(2,i);
    n(2,i)=v(1,i);
end

flag = 1;

for i = 1:4;
    if n(1:2,i)'*(x-P(1:2,i)) > 0;
        flag = 0;
        break;
    end
end
```

Prufum að keyra þetta og dæla inn ferhyrning úr `square` og núllpunktinum:



Mynd 2: Ferhyrningurinn sem var teiknaður (sjá má að núllpunkturinn er innan hans)

Úttak af skipanalínu MATLAB:

```
>> square_check([0; 0],square(-1,1,-1,1))
:
ans =

     1
```

3

Þetta forrit er forritið sem gerir mestu vinnuna, en það er forritið sem finnur fyrir okkur, með aðferð newtons, stöðupunkta fallsins. Það tekur inn f:fallið sem á að athuga, epsilon: sem tilgreinir skekkjuna frá réttum punkti sem við sættum okkur við, Δ sem ákvarðar hversu lítill stigullinn má vera, það er hversu nákvæmlega þetta er stöðupunktur og hversu nálægt Jacobi fylkið sem við erum að athuga er frá því að vera óandhverfanlegt, en einnig hversu margar ítranir við viljum keyra, til að koma í veg fyrir að við lendum í óendanlegri lykkju. Einnig má það taka inn endapunkta hnitakerfisins, en það er notað til þess að reikna út h1, sem notuð er við nákvæmni í útreiknun á stigli og Jacobi fylki fallsins.

```
function x = newton_gradient(f,epsilon, △, nmax, x0, P, axis)
   %Skgr. follin sem vid notum
   a =0; b = 1; c = 0; d = 1;
   if(nargin == 7)
       a = axis(1);
       b = axis(2);
       c = axis(3);
       d = axis(4);
   end

   h1 = 0.01 * min(b-a,d-c);
   F = @(x) Fgeneral(f,h1,x);
   dF = @(x) dFgeneral(f,h1,x);
   % Upphafsstilling:
   n=0;
   x=x0;
   %fprintf('%1d  %21.15e\n',n,x')
   y=F(x);
   dy=dF(x);

   %Possum ad fylkid se innan reiknimarka andhverfanlegt, haettum
   %ef svo er ekki
   if abs(det(dy)) < △
       return
   end
   h=-dy\y;
   x=x+h;
   n=1;
   e0=norm(h);
   e=2*epsilon;
   while e>epsilon && norm(y)>△ && n<nmax && square_check(x,P)
       y=F(x); dy=dF(x);
       if abs(det(dy)) < △
           return
       end
       h=-dy\y;
       e=norm(h);
       x=x+h;
       n=n+1;
       e0=e;
   end
end
```

Keyrsla með nokkrum dæmum. ϵ, δ voru valin nógu lítil og N nógu stórt þannig að forritið tæki ekki of langan tíma, en hinsvegar skilaði frekar nákvæmum niðurstöðum:

```
P = [ -1,-1,1,1,-1;-1,1,1,-1,-1];
f = @(x,y) x^2 + y^2;
newton_gradient(f,0.001,0.001,100,[0.5;0.5],P)
f = @(x,y) x*y;
newton_gradient(f,0.001,0.001,100,[0.5;0.5],P)
f = @(x,y) sin(x) + cos(y);
newton_gradient(f,0.001,0.001,100,[-1.5;0.0],2*P)
```

```
ans =

     0
     0
```

```



ans =

     0
     0


ans =

  -1.570796327777076
                   0
```

4 Handstýrð leit að stöðupunktum

Nú erum við loks komin á þann stað að geta farið að leita að stöðupunktum. Forrit þetta teiknar fyrst upp jafnhæðarferla gefins falls og gefur svo kost á því að velja ferhyrning og svo byrjunarpunkt. Þá notar forritið `newton_gradient` til að leita að stöðupunkti. Ef hann finnst er hann flokkaður, staðsetning og gildi hans skrifað út á úttakslínu og hann er merktur inn á myndina. Ef hann hann finnst ekki gerist ekkert. Svo er gefinn kostur á því að halda áfram og finna nýja stöðupunkta. Forritið hættir þegar smellt er á hægri músarhnapp.

Fallið f úr lið i

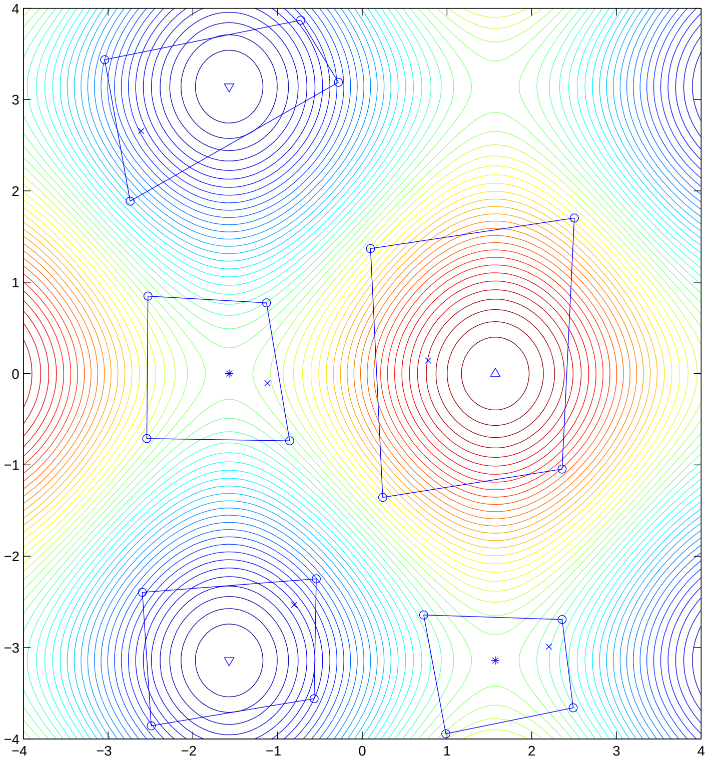
```
function r = func(x,y)
  r = sin(x) + cos(y);
end
```

Forritið

Athugið að mcode virðist ekki styðja íslenska stafi, en útprentað er “Hápunktur í”, “Lágpunktur í” og “Söðulpunktur í”, ef punktur finnst, en “Ekki er hægt að segja til um (x,y) =” ef ekki er hægt að segja til um punktinn.

```
function manualCriticalPointSearch(f,axiss,epsilon,△,nmax)
    a=axiss(1);
    b=axiss(2);
    c=axiss(3);
    d=axiss(4);

    axis(axiss);

    x = linspace(a,b,250);
    y = linspace(c,d,250);

    [X,Y] = meshgrid(x,y);

    Z = arrayfun(f,X,Y);

    clf;
    contour(X,Y,Z,50)

    hold on
    hnappur = 1;

    while hnappur == 1
        P = square(a,b,c,d);
```

```
        %Vid faum P = [] ef vid haettum i midri keyrslu, t.d. ef
        %smellt er a haegri hnapp.
        if length(P) == 0
            return
        end

        [x, y, hnappur] = ginput(1);
        if hnappur ≠ 1
            return
        end
        plot(x,y,'x');
        x0 = [ x; y];

        while (square_check(x0,P) ≠ 1)
            [x, y, hnappur] = ginput(1);
            plot(x,y,'x');
            x0 = [ x; y];
        end


        p = newton_gradient(f,epsilon,△,nmax,x0,P,axiss);
        if(square_check(p,P))
            h1 = 0.01 * min(b−a,d−c);
            Hessian = dFgeneral(f,h1,p);
            M = det(Hessian);

            %Ef M er svona litid, tha er determinant fylkisins
            %ansi nalgaegt thvi ad vera 0, og er thvi ekki
            %haegt ad not thad i reikningum. Tha er heldur ekki
            %haegt ad segja neitt um thann punkt, thannig ad
            %vid sleppum honum bara
            if abs(M) < epsilon
                fprintf('Ekki h gt  a   segja til um (x,y) = (%f,%f)\n',p(1),p(2))
                plot(p(1),p(2),'o')
                continue
            end

            if(M > 0)
                eigs = eig(Hessian);
                if eigs(1) > 0 && eigs(2) > 0
                    fprintf('L gpunktur    (x,y) = (%f,%f), f(x,y) = ...
                        %f\n',p(1),p(2),f(p(1),p(2)))
                    plot(p(1),p(2),'v')
                else
                    if eigs(1) < 0 && eigs(2) < 0
                        fprintf('H punktur    (x,y) = (%f,%f), f(x,y) = ...
                            %f\n',p(1),p(2),f(p(1),p(2)))
                        plot(p(1),p(2),'^')
                    end
                end
            else
                if M < 0
                    fprintf('S ulpunktur    (x,y) = (%f,%f)\n',p(1),p(2))
                    plot(p(1),p(2),'*')
                end
            end
       else
            fprintf('Enginn punktur fannst innan kassans\n');
        end
    end

end
```



Mynd 3: Keyrsla á manual search

Keyrsluskrá:

```
f = @(x,y) sin(x) + cos(y);
manualCriticalPointSearch(f,[−4,4,−4,4],0.001,0.001,100)
```

Keyrsla, mynd má sjá á mynd 3.

```
H punktur    (x,y) = (1.570796,−0.000000), f(x,y) = 2.000000
S  ulpunktur    (x,y) = (−1.570796,0.000000)
L gpunktur    (x,y) = (−1.570796,3.141593), f(x,y) = −2.000000
L gpunktur    (x,y) = (−1.570796,−3.141593), f(x,y) = −2.000000
S  ulpunktur    (x,y) = (1.570796,−3.141593)
```

Forrit sem skilgreinir

$$f(x) = \sum_{j=1}^{N} \alpha_j e^{-\frac{||x-q_j||^2}{\epsilon}}$$

```
function y =  func2(alphas, qs, eps, x)
  y= 0;
```

```
  for i= 1:length(alphas)
    y = y+ alphas(i)* exp(-(dot((x-qs(i,:)), (x-qs(i,:))))/eps);
  end
end
```

Einnig er wrapper fyrir fallið, sem einfaldar notkun þess í forritinu. Athugið að ekki er hægt að senda fall inn í fall úr commandline í matlab nema að það sé anonymous.

```
function k = func2wrapper(inx,iny)
  eps = 0.04;
  a = [1,-1,1,-1];
  q = [0.5 0.5; 0.5 -0.5; -0.5 -0.5; -0.5 0.5];
  x = [inx,iny];
  k = func2(a,q,eps,x);
end
```

Forritið mætti vinna áfram og gera það t.d. sjálfvirkt, þannig mætti forða manni frá því að vera að tékka handvirkt, og auðvelda manni þannig vinnuna. Einnig væri kannski sniðugt að láta það plotta í þrívídd, og sýna manni þannig að hágildi og lággildi eru í raun í punktunum, þ.e. að þar eru hápunktar og lágpunktar sléttunnar.

5 Aukaliður

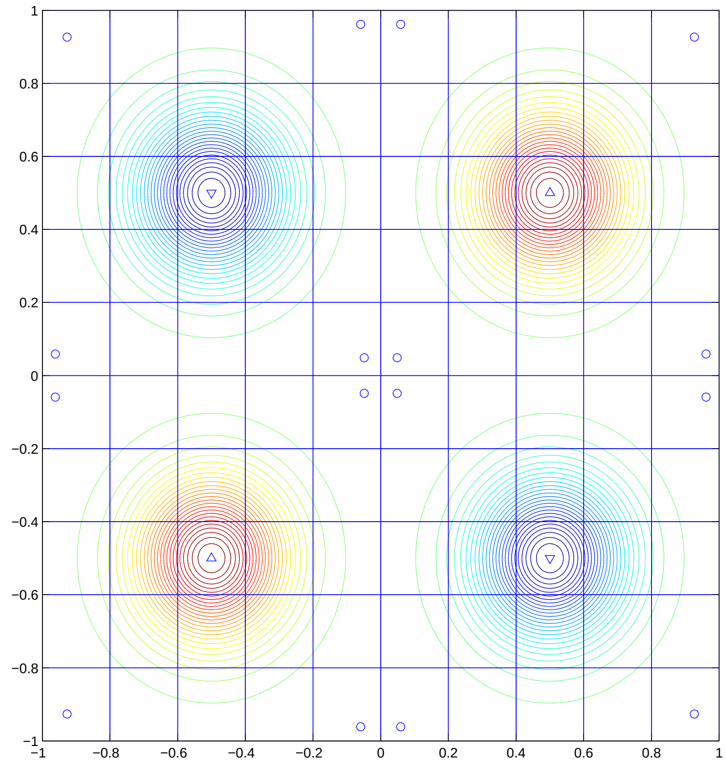
Fall sem skiptir svæðinu öllu í reiti og leitar kerfisbundið að stöðupunktum Fallið er keyrt á hliðstæðan hátt og handstýrða stöðupunktaleitin nema hvað bæta þarf inn fallgildinu boxes sem segir til um hve margar skiptingar á að gera á svæðinu á hvorn ás svo svæðinu er skipt í $boxes^2$ reiti og leitar innan hvers og eins reitar að sérstöðupunktum og flokkar þá sem hægt er.

```
function automaticCriticalPointSearch(f,axiss,epsilon,△,nmax,boxes)
    a=axiss(1);
    b=axiss(2);
    c=axiss(3);
    d=axiss(4);

    axis(axiss);

    xspace = linspace(a,b,250);
    yspace = linspace(c,d,250);

    [X,Y] = meshgrid(xspace,yspace);


    Z = arrayfun(f,X,Y);

    clf;
    contour(X,Y,Z,50)
    hold on

    i = 1;
    j = 1;

    x = linspace(a,b,boxes);
    y = linspace(c,d,boxes);

    while i < boxes
       while j < boxes
            P = [ x(i) x(i) x(i+1) x(i+1) x(i); ...
                  y(j) y(j+1) y(j+1) y(j) y(j)];
            plot(P(1,:),P(2,:))
        x0 = [ (x(i) + x(i+1))*0.5; (y(j) + y(j+1))*0.5];
            try
```

```
                p = newton_gradient(f,epsilon,△,nmax,x0,P, ...
                                    axiss);
            catch err
                continue
            end

        if(square_check(p,P))
        h1 = 0.01 * min(b−a,d−c);
        Hessian = dFgeneral(f,h1,p);
        M = det(Hessian);

                %Ef M er svona litid, tha er determinant fylkisins
                %ansi nalgaegt thvi ad vera 0, og er thvi ekki
                %haegt ad not thad i reikningum. Tha er heldur ekki
                %haegt ad segja neitt um thann punkt, thannig ad
                %vid sleppum honum bara
                if abs(M) < epsilon
                    j = j+1;
                    fprintf('Ekki h gt  a   segja til um (x,y) = (%f,%f)\n',p(1),p(2))
                    plot(p(1),p(2),'o')
                    continue
                end

        if(M > 0)
            eigs = eig(Hessian);
            if eigs(1) > 0 && eigs(2) > 0
                        fprintf('L gpunktur    (x,y) = (%f,%f), f(x,y) = ...
                            %f\n',p(1),p(2),f(p(1),p(2)))
                        plot(p(1),p(2),'v')
                    else
                        if eigs(1) < 0 && eigs(2) < 0
                            fprintf('H punktur    (x,y) = (%f,%f), f(x,y) = ...
                                %f\n',p(1),p(2),f(p(1),p(2)))
                            plot(p(1),p(2),'^')
                        end
            end
        else
            if M < 0
            fprintf('S ulpunktur    (x,y) = (%f,%f)\n',p(1),p(2))
            plot(p(1),p(2),'*')
            end
        end
        end
        j = j+1;
    end
    i = i +1;
        j = 1;
    end
end
```

Keyrsluskrá acpkeyrsla1:

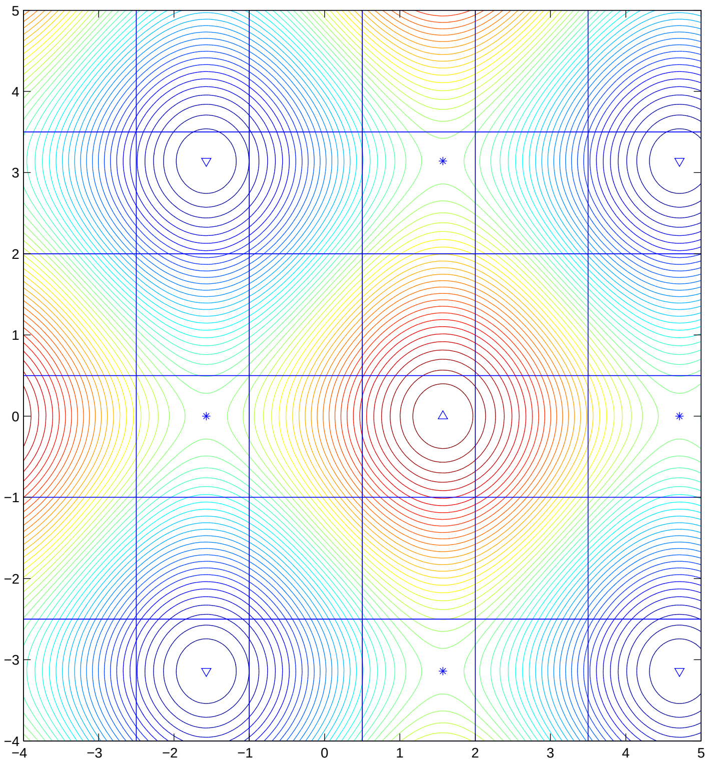
```
f = @(x,y) func2wrapper(x,y);
automaticCriticalPointSearch(f,[−1,1,−1,1],0.001,0.001,100,11)
```



Mynd 4: Keyrsla á acpkeyrsla1

Úttak keyrslu:

```
Ekki h gt  a   segja til um (x,y) = (−0.926579,−0.926579)
Ekki h gt  a   segja til um (x,y) = (−0.961572,−0.058996)
Ekki h gt  a   segja til um (x,y) = (−0.961572,0.058996)
Ekki h gt  a   segja til um (x,y) = (−0.926579,0.926579)
H punktur    (x,y) = (−0.500000,−0.500000), f(x,y) = 1.000000
L gpunktur    (x,y) = (−0.500000,0.500000), f(x,y) = −1.000000
Ekki h gt  a   segja til um (x,y) = (−0.058996,−0.961572)
Ekki h gt  a   segja til um (x,y) = (−0.048544,−0.048544)
Ekki h gt  a   segja til um (x,y) = (−0.048544,0.048544)
Ekki h gt  a   segja til um (x,y) = (−0.058996,0.961572)
Ekki h gt  a   segja til um (x,y) = (0.058996,−0.961572)
Ekki h gt  a   segja til um (x,y) = (0.048544,−0.048544)
Ekki h gt  a   segja til um (x,y) = (0.048544,0.048544)
Ekki h gt  a   segja til um (x,y) = (0.058996,0.961572)
L gpunktur    (x,y) = (0.500000,−0.500000), f(x,y) = −1.000000
H punktur    (x,y) = (0.500000,0.500000), f(x,y) = 1.000000
Ekki h gt  a   segja til um (x,y) = (0.926579,−0.926579)
Ekki h gt  a   segja til um (x,y) = (0.961572,−0.058996)
Ekki h gt  a   segja til um (x,y) = (0.961572,0.058996)
Ekki h gt  a   segja til um (x,y) = (0.926579,0.926579)
```



Mynd 5: Keyrsla á acpkeyrsla2

Keyrsluskrá acpkeyrsla2:

```
f = @(x,y) sin(x) + cos(y);
automaticCriticalPointSearch(f,[−4,5,−4,5],0.001,0.001,100,7)
```

Úttak keyrslu:

```
L gpunktur    (x,y) = (−1.570796,−3.141593), f(x,y) = −2.000000
S ulpunktur    (x,y) = (−1.570796,0.000000)
L gpunktur    (x,y) = (−1.570796,3.141593), f(x,y) = −2.000000
S ulpunktur    (x,y) = (1.570796,−3.141593)
H punktur    (x,y) = (1.570796,0.000000), f(x,y) = 2.000000
S ulpunktur    (x,y) = (1.570796,3.141593)
L gpunktur    (x,y) = (4.712389,−3.141593), f(x,y) = −2.000000
S ulpunktur    (x,y) = (4.712389,0.000000)
L gpunktur    (x,y) = (4.712389,3.141593), f(x,y) = −2.000000
```

Að skýrsluni unnu : ____________________ ____________________ ____________________