
```
%
% CA3_SM.m -- dhp -- 29 jan 2019
%

clear

% hi function

% define domain in x and y
xx = -2.5:0.025:2.5; yy = -2.5:0.025:2.5;

% define a 'mesh' to plot on
[xg,yg] = meshgrid(xx,yy);

% set root-finding tolerance (for initial pt & loop)
tol = 1e-10;
fzero_opt = optimset('TolX',tol);

% root-finding loop control parameters
%ds = 0.6;
ds = 0.1;
% useful future variables
itmax = 24;
%delta = pi/2;
delta = pi/2;

% define the function HI(x,y)
hi = @(x,y) exp(-3*((x + 0.5).^2 + 2*y.^2)) + exp(-x.^2 -
    2*y.^2).*cos(4*x) - 1e-3;

% define the function HI on the circle (radius = ds)
hi_th = @(th,xn,yn) hi(xn + ds*cos(th),yn + ds*sin(th));

% find point on the "H" with y=0
% initial guess for a point very NEAR contour
xi = -1.97; yi = 0;

% START: FIND INITIAL POINT on contour (you can use fzero here)
% root-find angle to point ON contour
th = 0;
th = fzero(@(th) hi_th(th,xi,yi),th,fzero_opt);
%
% END : FIND INITIAL POINT on contour

% compute first point ON contour
xn = xi + ds*cos(th);
yn = yi + ds*sin(th);

% make array of contour points
Nsteps = 24000;
zero_contour = zeros(Nsteps+1,2);
zero_contour(1,:) = [xn yn];
```

```

total_evals = 0;

% loop for the contour
for kk = 1:Nsteps
    % START: theta root-finding here (you cannot use fzero here!!)
    %
    % Secant Method for next point

    % 0) set two initial guesses
    th0 = th-delta; hi_th0 = hi_th(th0,xn,yn);
    thn = th+delta;
    Nevals = 1;

    check = thn-th0;

    % root-finding loop
    while( abs(check) > tol )
        if Nevals > itmax
            fprintf('no convergence:\n')
            fprintf('\tds = %f, delta = %f\n', ds, delta)
            break
        end
        % 1) function evaluation at thn
        hi_thn = hi_th(thn, xn,yn);
        Nevals = Nevals + 1;

        % 2) secant update
        thS = thn - (hi_thn * (thn - th0)/(hi_thn - hi_th0));

        % 3) prepare next iteration
        th0 = thn; hi_th0 = hi_thn;
        thn = thS;

        check = thn - th0;
    end

    total_evals = total_evals + Nevals;

    %
    % END: theta root-finding here
    if Nevals > itmax
        break
    end

    % Compute next point on contour
    xn = xn + ds*cos(thn);
    yn = yn + ds*sin(thn);

    % update new points & angle
    zero_contour(kk+1,:) = [xn yn];
    th = thn;
end

```

```
avg_evals = total_evals / kk

% colour contourplot of HI function
figure(2); clf
pcolor(xx,yy,hi(xg,yg)); colorbar
shading interp; hold on
contour(xx,yy,hi(xg,yg),[0 0], 'w--')
axis equal; axis image

title('trace the contour hi(x,y)=0')
xlabel('x-axis')
ylabel('y-axis')

% plot the zero-contour, 1st & last point
plot(zero_contour(:,1),zero_contour(:,2), 'ro-')
plot(zero_contour(1,1),zero_contour(1,2), 'ko')
plot(zero_contour(end,1),zero_contour(end,2), 'k*')
```

Published with MATLAB® R2018a