DOING PHYSICS WITH MATLAB

MATHEMATICAL ROUTINES

TURNING POINTS OF A FUNCTION

Ian Cooper
School of Physics, University of Sydney
ian.cooper@sydney.edu.au

DOWNLOAD DIRECTORY FOR MATLAB SCRIPTS

turningPoints.m

Matlab function to determine the turning points of a function. The function outputs in a Figure Window a plot of the function showing the maxima and minima points and displays in the Command Window the indices for the maxima and minima, and the x and y values at these points.

You are often required to find the stationary points of a curve where the gradient of the curve is zero. The function **turningPoints.m** can be used to find the stationary points corresponding to points of maxima and minima of a curve.

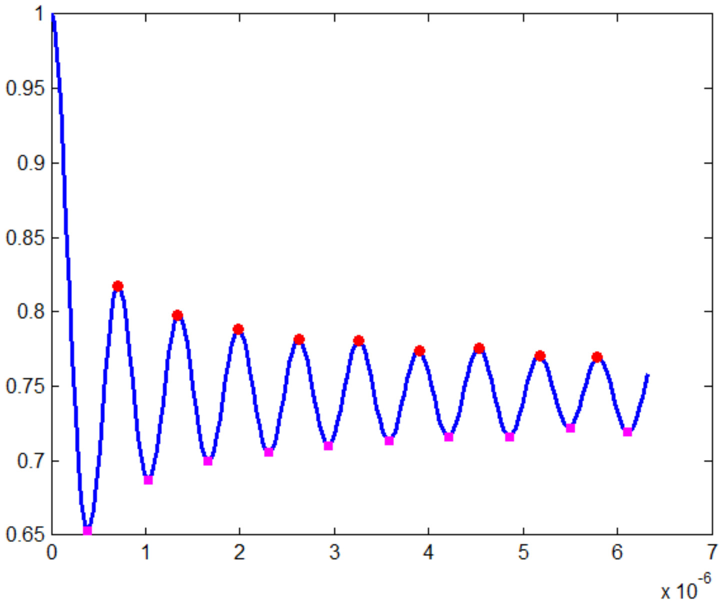
The x and y data for the curve are passed to the function in the Command window.

For example, turningPoints(xP,WP) is entered into the Command Window to pass on the variables xP (x data) and WP (y data) to the function turningPoints.m.

```
function turningPoints(xData, yData)
```

Sample results

Figure Window output: the maxima are marked as red dots and the minima are marked by magenta squares.



The indices and the x and y values are displayed in the Command Window.

```
          Max values - indices / xData / yData

          indexMax =   23    43    63    83   103   123   143   163   182

          ans =      1.0e-05 *

   0.0703    0.1342    0.1981    0.2621    0.3260    0.3899    0.4538    0.5177    0.5785

          ans =

   0.8174    0.7978    0.7886    0.7816    0.7804    0.7740    0.7760    0.7705    0.7693


          Min values - indices / xData / yData

          indexMin =    13    33    53    73    93   113   133   153   173   192

ans =   1.0e-05 *

  0.0384    0.1023    0.1662    0.2301    0.2940    0.3579    0.4219    0.4858    0.5497
  0.6104

ans =

    0.6527    0.6873    0.6998    0.7055    0.7104    0.7138    0.7160    0.7164    0.7223
    0.7190
```

mscript

```
function turningPoints(xData, yData)

size = length(xData);     %Get the length of the dataset

a1 = yData(1,1); a2 = yData(1,2);
if a2 > a1, flag = 1; else flag = 2; end;
v = 0;

% find max
for x = 2:size-1
    a1 = yData(1,x);      %Get two adjacent samples of the dataset
    a2 = yData(1,x+1);

    if flag == 1 && a2 > a1; x = x+1; end;
    if (flag == 1 && a2 < a1); v = v + 1; indexMax(v) = x; x = x+1;
end;
    if a2 <= a1, flag = 0; end;
    if a2 > a1, flag = 1; end;
end

a1 = yData(1,1); a2 = yData(1,2);
if a2 < a1, flag = 1; else flag = 2; end;
v = 0;

% find min
for x = 2:size-1
    a1 = yData(1,x);      %Get two adjacent samples of the dataset
    a2 = yData(1,x+1);

    if flag == 1 && a2 < a1; x = x+1;
    end;
    if (flag == 1 && a2 > a1); v = v + 1; indexMin(v) = x; x = x+1;
end;
    if a2 >= a1, flag = 0; end;
    if a2 < a1, flag = 1; end;
end

figure(99)
plot(xData,yData,'lineWidth',2)
hold on

hp1 = plot(xData(indexMax),yData(indexMax),'o');
set(hp1,'MarkerEdgeColor','r','MarkerFaceColor','r','MarkerSize',5);

hp1 = plot(xData(indexMin),yData(indexMin),'s');
set(hp1,'MarkerEdgeColor','m','MarkerFaceColor','m','MarkerSize',4);

disp('  ')
disp('Max values - indices / xData / yData')
indexMax
xData(indexMax)
yData(indexMax)

disp('  ')
disp('  ')
disp('Min values - indices / xData / yData')
indexMin
xData(indexMin)
yData(indexMin)
```