

```

1 function [ mVorticity , sCor , excluded ] = getVortexParam(filename , pathname , num , s)
2
3 if ( nargin < 3)
4     num = 2;
5 end
6
7 if ( nargin < 4)
8     s = 1;
9 end
10
11 fprintf( 'getVortexParam : Appel de followVortex \n');
12
13 [ vortex , vorticity , cX , cY , excluded ] = followVortex(filename , pathname , num , s);
14 sCor = {};
15 mVorticity = {};
16
17 fprintf( 'getVortexParam : Extraction des vorticités moyennes et des abscisses
18         curvilignes \n')
tic;
19
20 for i = 1:s
21     smCy = smooth(cY{1,i}(:, ) , 5);
22
23     xx = linspace(cX{1,i}(1) , cX{1,i}(end) , 1000);
24
25     try
26         yy = interp1(cX{1,i}(:, ) , smCy, xx);
27     catch
28         fprintf( 'getVortexParam : La suppression de tout un set de données n\'est
29             pas conseillée \n');
30         sCor{i} = {};
31         mVorticity{i} = {};
32         break;
33     end
34
35     figure(2);
36     hold on
37     plot(xx,yy);
38     hold off
39
40     XX = diff(xx);
41     YY = diff(yy);
42
43     S = (XX.^2 + YY.^2).^(1/2);
44     s = [0 cumsum(S)];
45
46     cXX = repmat(cX{1,i}(:, ) , 1 , length(xx));
47     cYY = repmat(cY{1,i}(:, ) , 1 , length(yy));
48
49     xxT = repmat(xx' , 1 , length(cX{1,i}(:, )));
50     yyT = repmat(yy' , 1 , length(cY{1,i}(:, )));
51
52     dXX = abs(xxT'-cXX);
53     dYY = abs(yyT'-cYY);
54
55     [M,I] = min(dXX');
56     sCor{i} = s(I);
57     mVorticity{i} = cellfun (@mean, vorticity(~cellfun (@isempty , vorticity(:, i))) , i));

```

```

58     try
59         figure(3);
60         hold on
61         plot(sCor{i}, mVorticity{i}, '.');
62         hold off
63
64     catch
65         fprintf('getVortexParam : Une erreur a été rencontré lors de l\'affichage,
66             données sauvegardées \n');
67     end
68
69     elapsed = toc;
70     fprintf('getVortexParam : Extraction achevée en %d secondes \n', ...
71             elapsed);
72 end

```