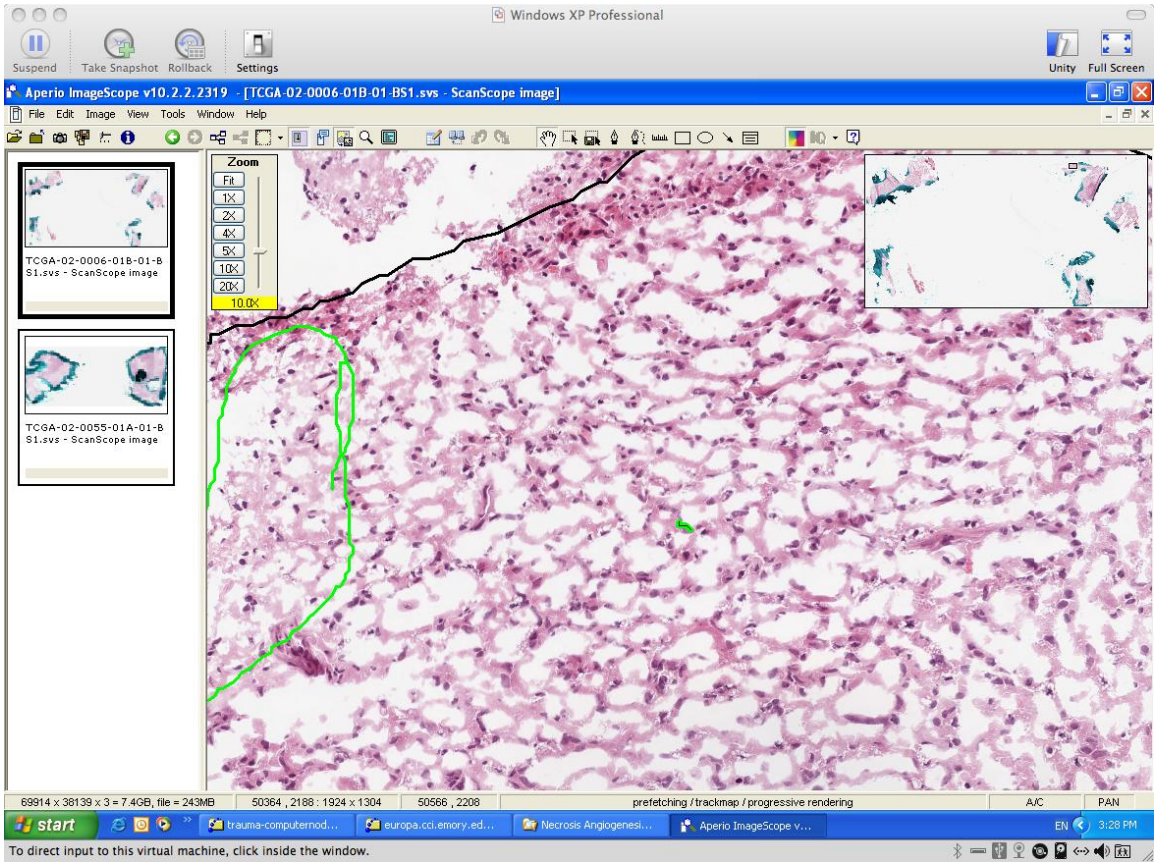
Markup Boundary Fixing for Spatial Databases

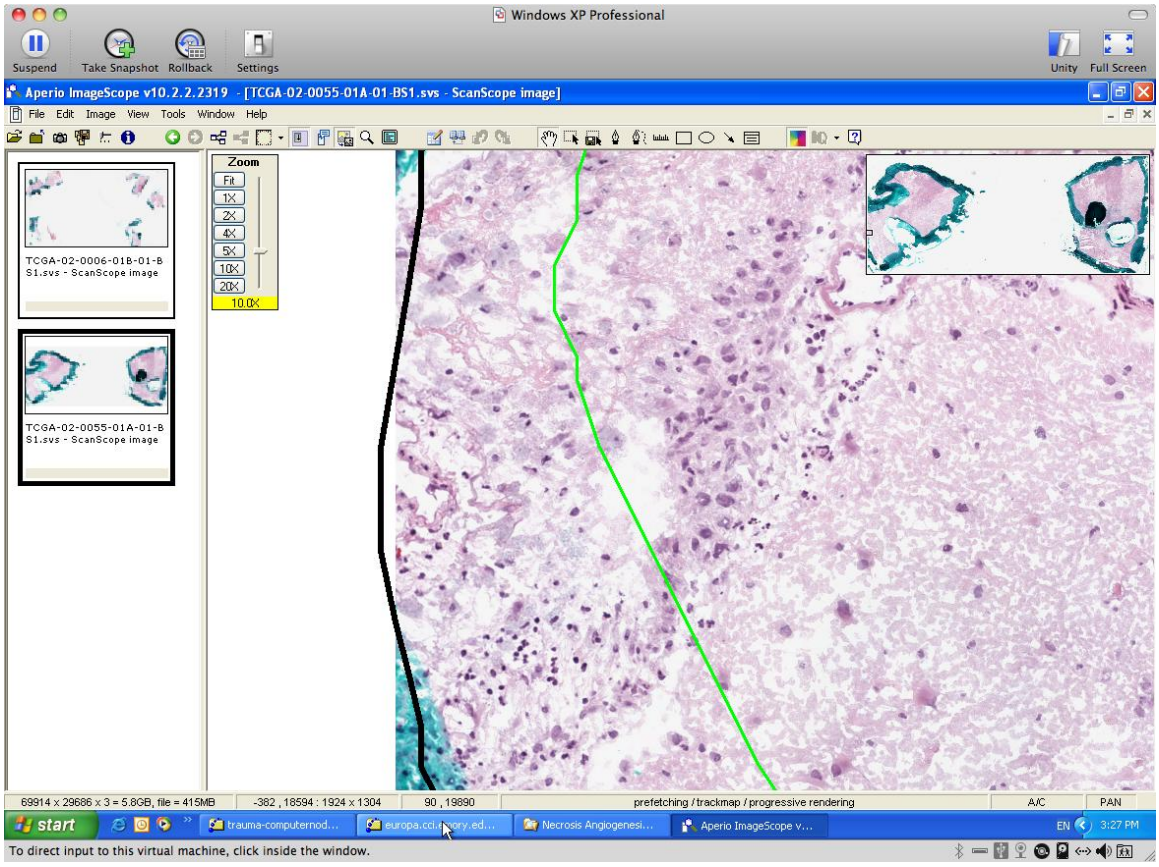
1 When **the contour length is less than 4 points**, it is doomed to fail. Given the fact that the first point is identical to the last, only one point is different from the first and last point. As a result, it is not a valid polygon.

Solution: When this occurs, we simply discard it.



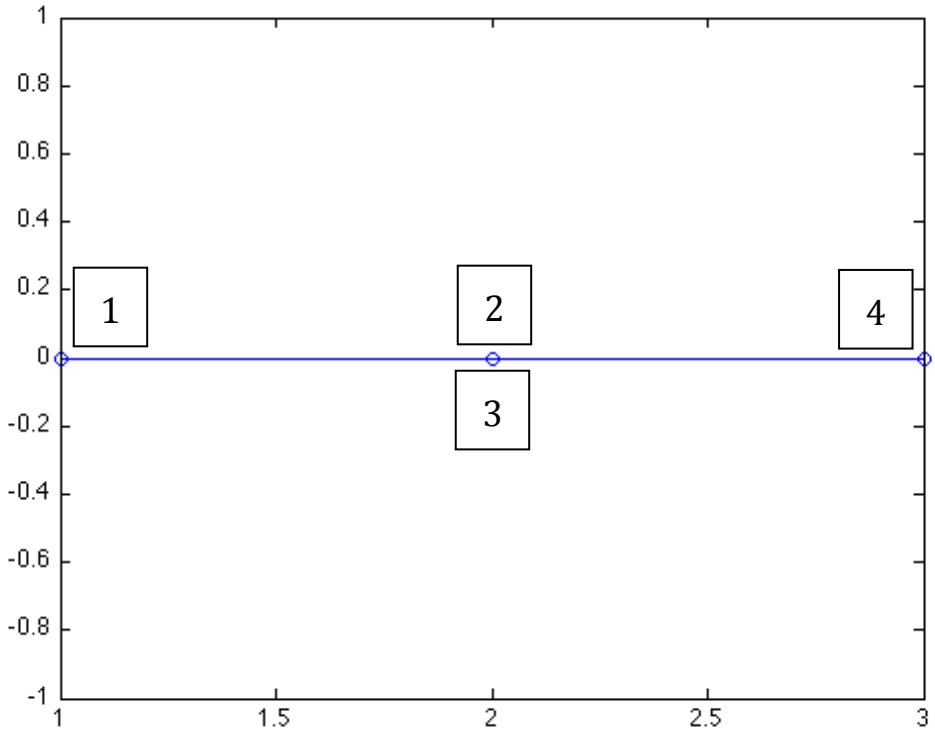
2 When **<u>negative points are in a contour</u>**, db2 doesn't accept that.

<u>Solution:</u> When this occurs, we remove points having negative coordinates.



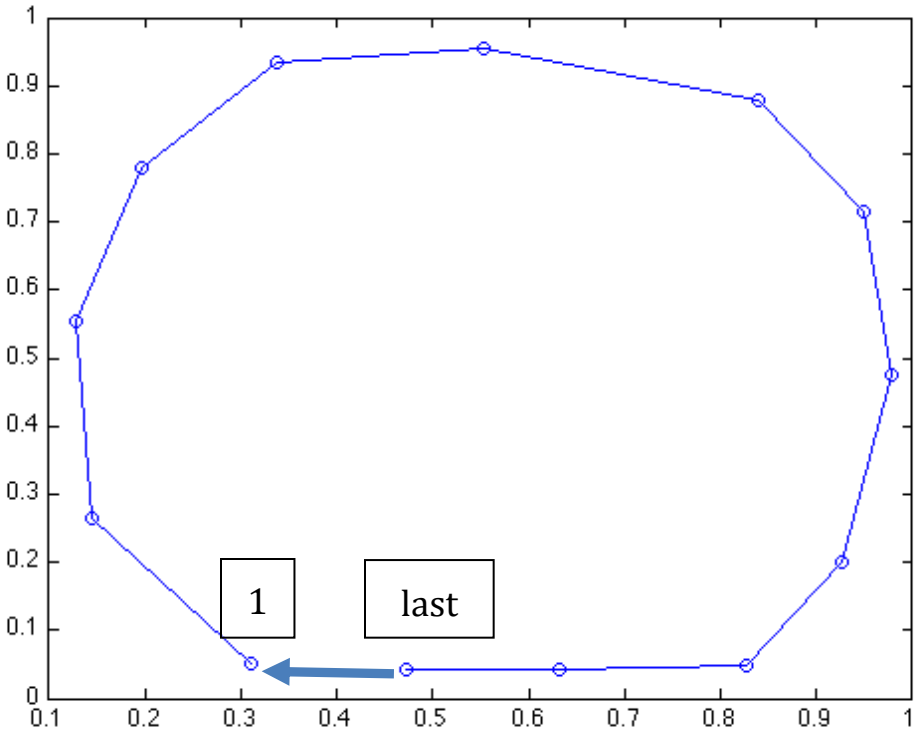
3 When **same points are repeated in a row in a contour**, db2 doesn't accept that.

Solution: Compute the distance of any pair of adjacent points and only keep one instance of identical points in a row.



4 When **a contour is open, i.e. the first point is not identical to the last point of the contour** , db2 doesn't accept that.

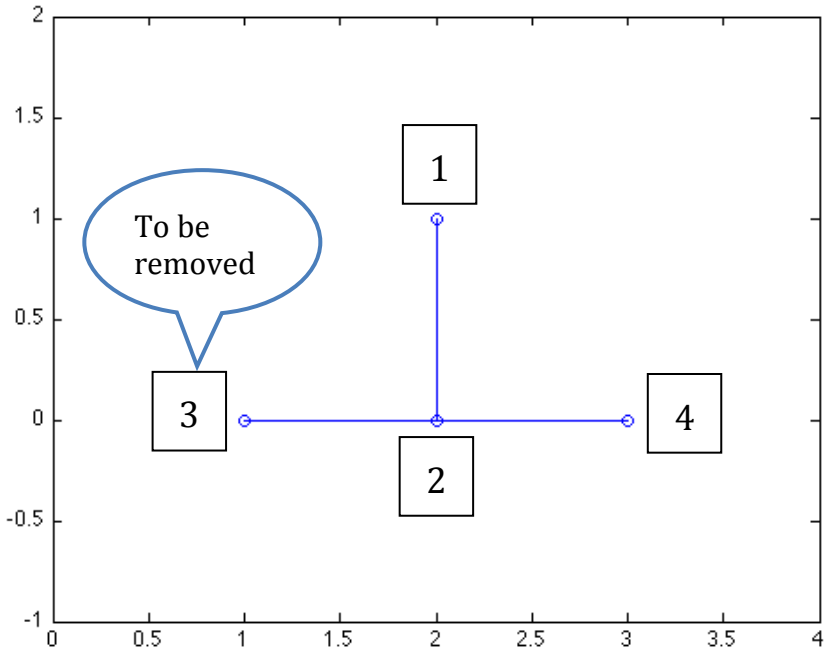
Solution: When this occurs, we duplicate the first point and append this duplicated point to the end of the contour



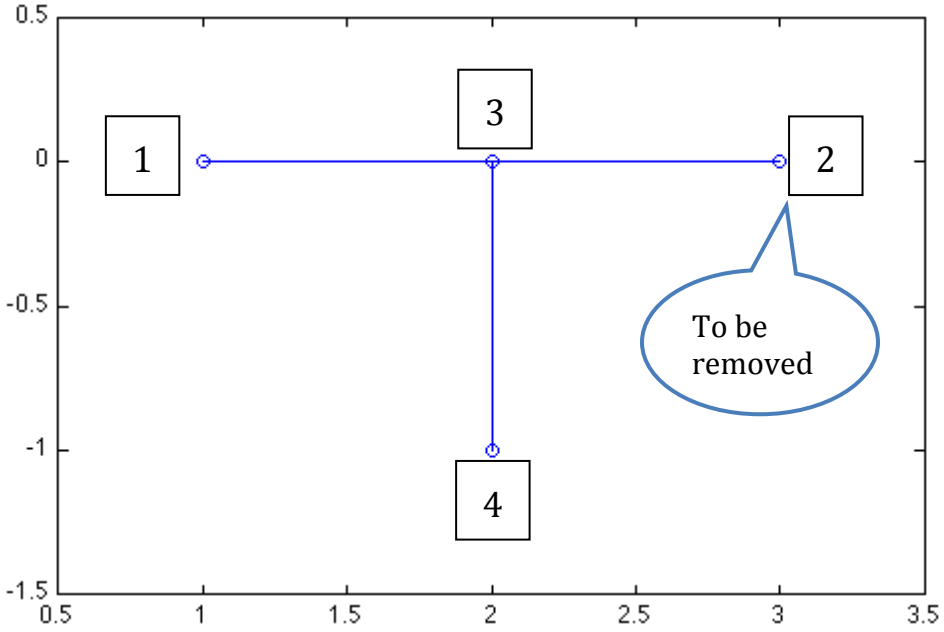
5 When **some points are on the segments defined by other points in the contour (T-conjunction)**, db2 doesn't accept that.

Solution: There are two T-conjunction sub-cases. Both of them can be detected by comparing slopes of adjacent segments and signs of changes in x- and y-direction. In either case, we remove the point connecting two adjacent segments with identical slope but different signs for changes in both x and y direction.

Case 1:

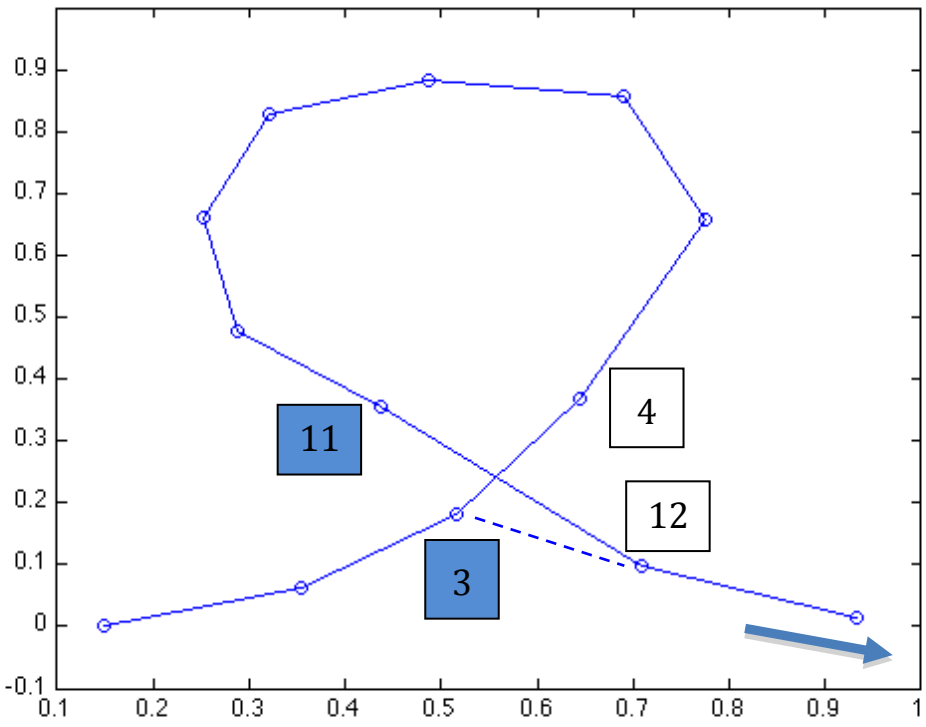


Case 2:

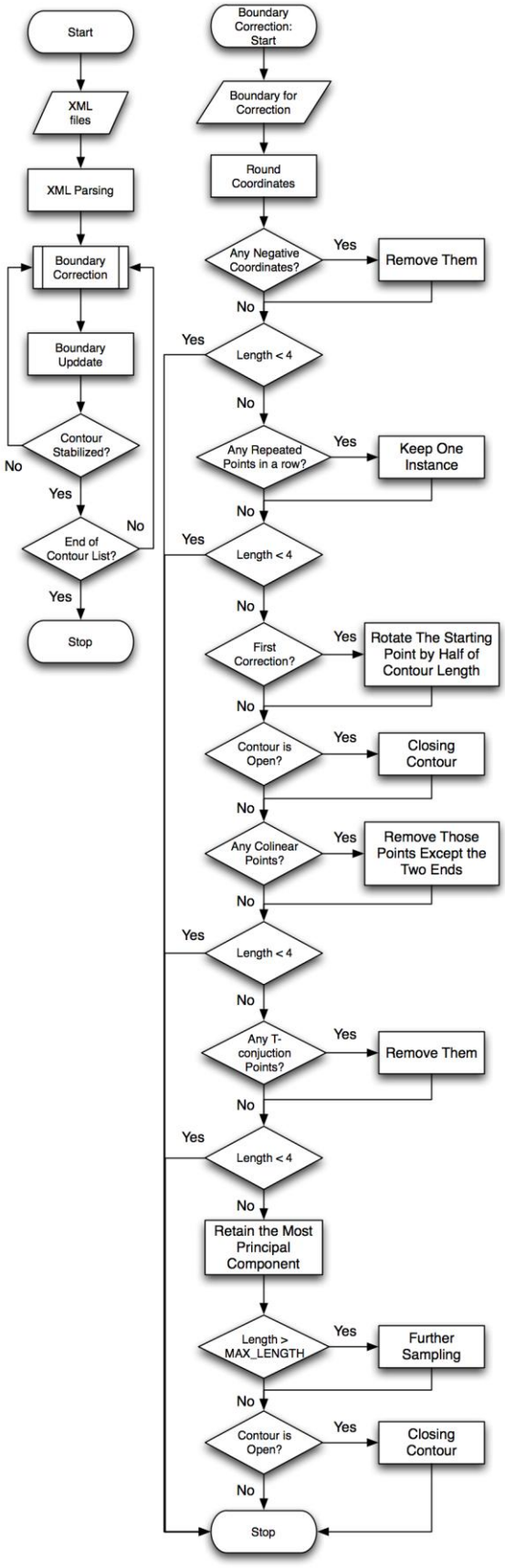


6 When **a segment of the contour goes across another segment of the contour, i.e. self-crossing**, db2 doesn't accept that.

Solution: Each segment is inspected to see if any other segment in the contour intersects it. The indices of the two intersected segments are recorded. The length of the self-crossing component (i.e. from point 4 to 11 in the example below) is compared with that of the remaining contour. The one having shorter length is removed.

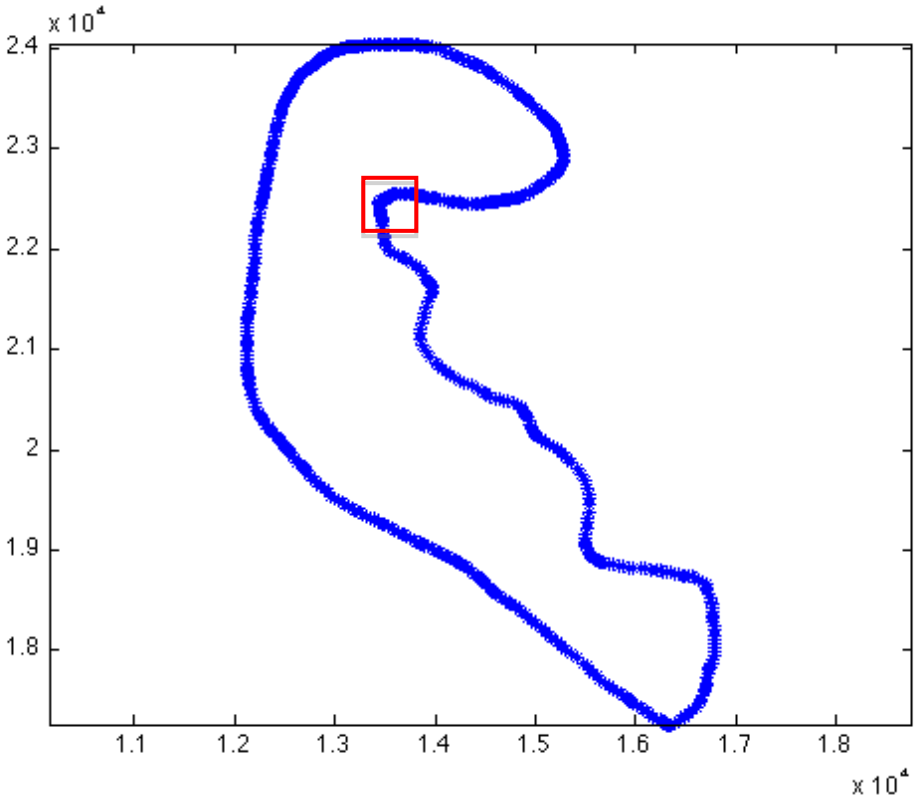


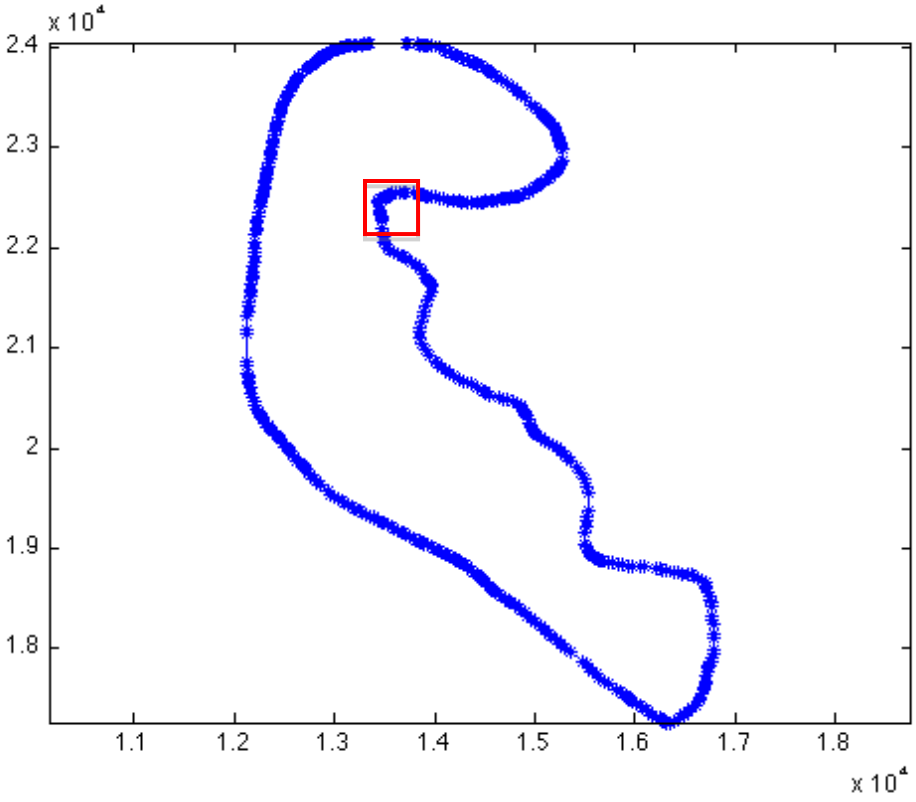
6 When any combination of the above six cases is occurred, db2 doesn't accept that.
Solution: We create a workflow to address this problem. Each input contour keeps being corrected until it is stabilized. The complete flowchart is given as follows.

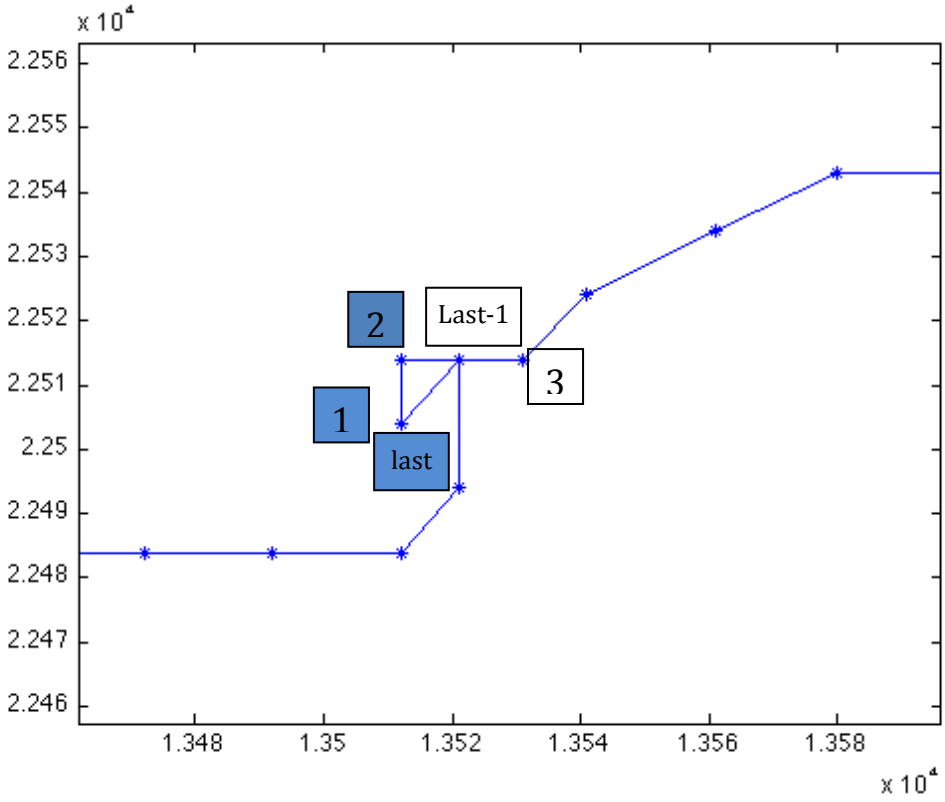


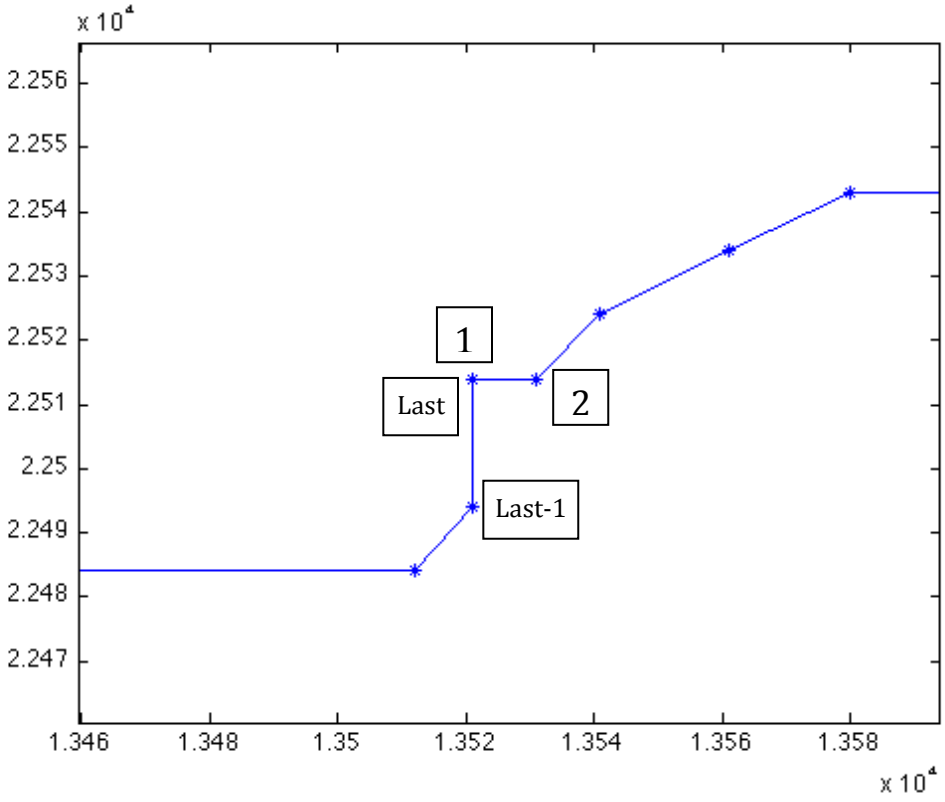
Following is a list of cases we encountered in processing our real data:

Case1:

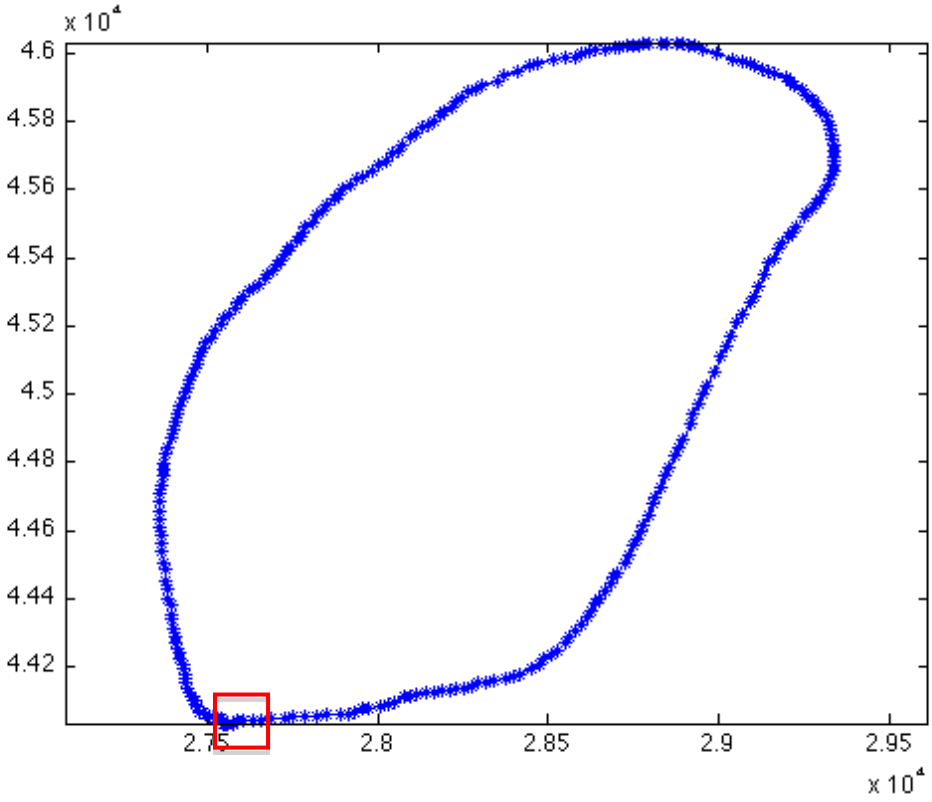


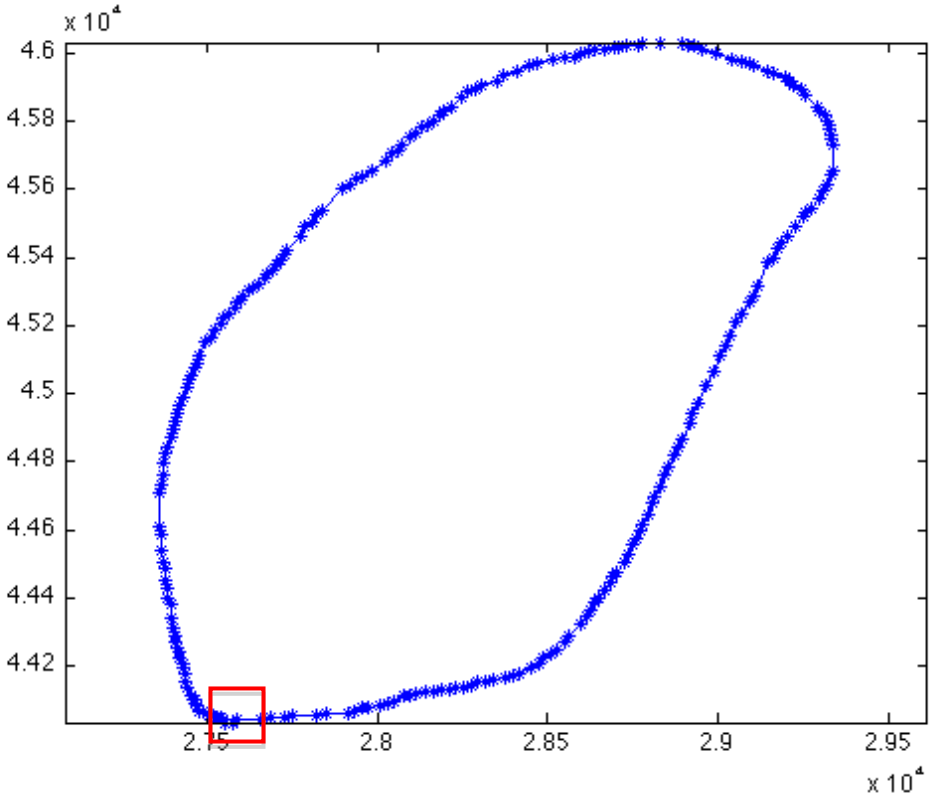


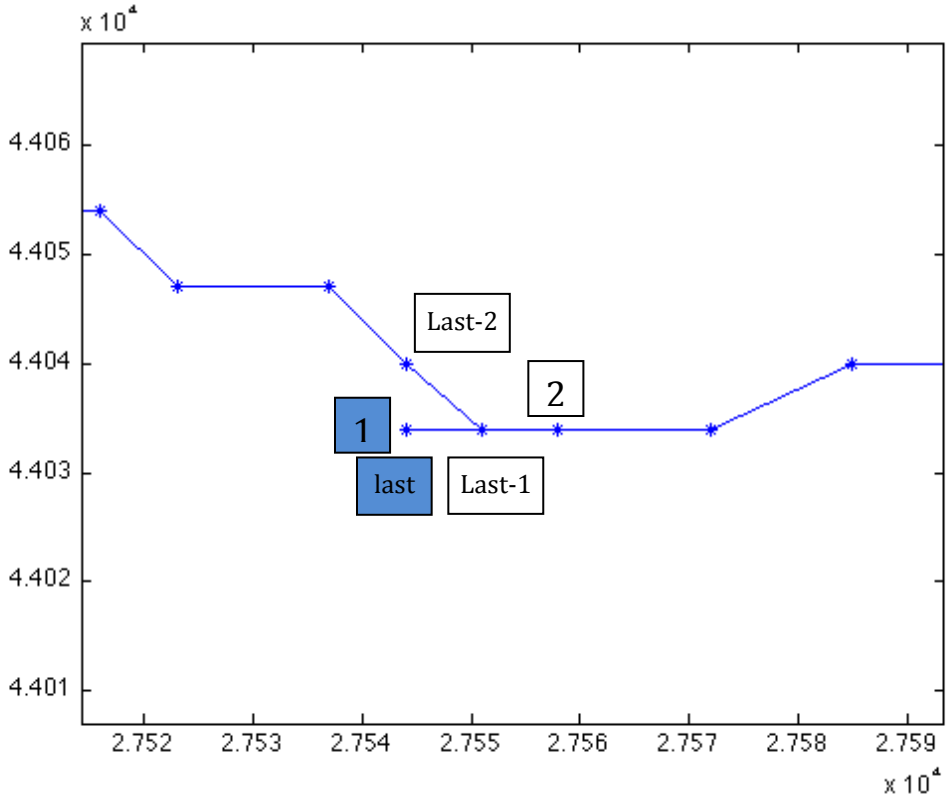

x 10^4
2.256
2.255
2.254
2.253
2.252
2.251
2.25
2.249
2.248
2.247
2.246
1.348
1.35
1.352
1.354
1.356
1.358
x 10^4
2
Last-1
3
1
last

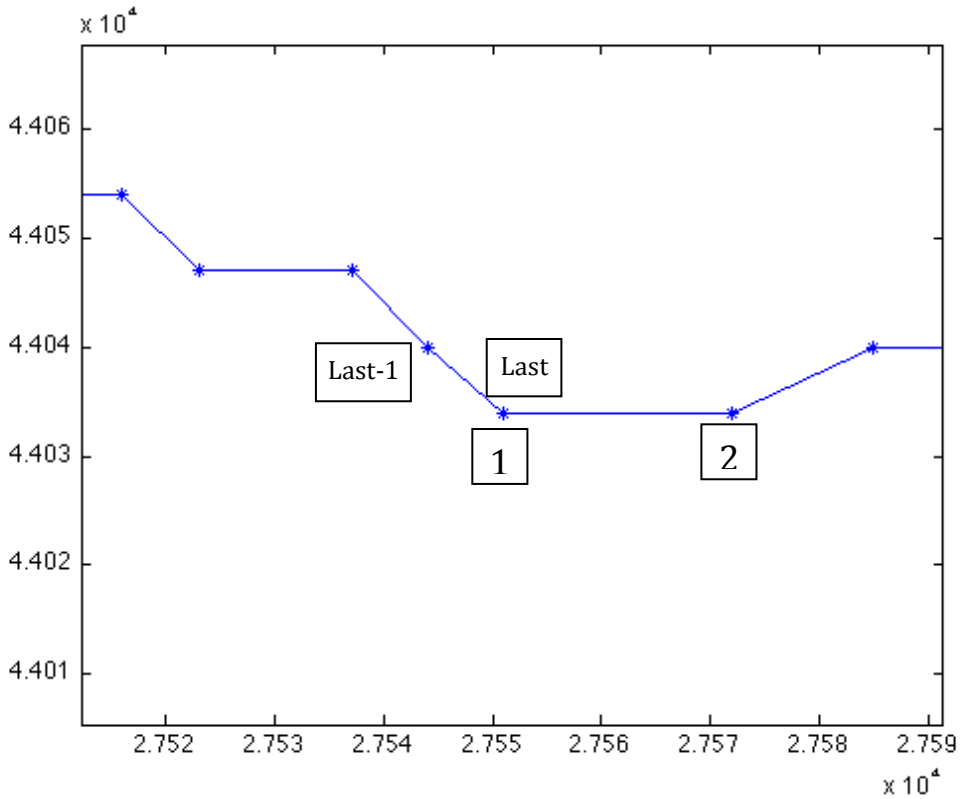

x 10^4
2.256
2.255
2.254
2.253
2.252
2.251
2.25
2.249
2.248
2.247
1.346
1.348
1.35
1.352
1.354
1.356
1.358
x 10^4
1
Last
2
Last-1

Case 2:

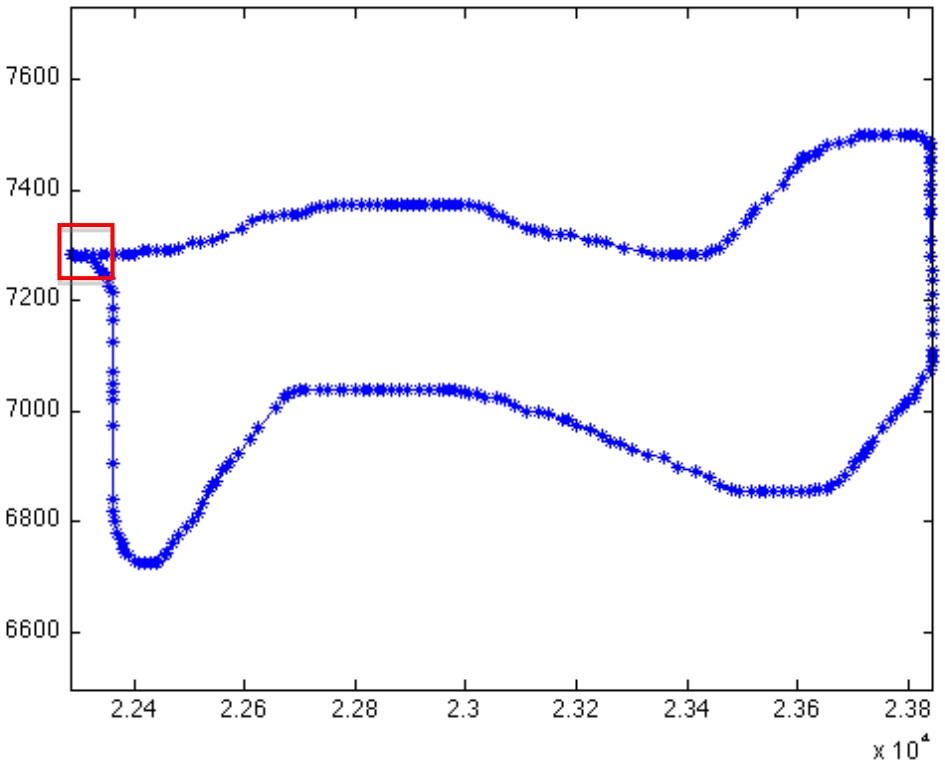

x 10^4
4.6
4.58
4.56
4.54
4.52
4.5
4.48
4.46
4.44
4.42
2.75
2.8
2.85
2.9
2.95
x 10^4

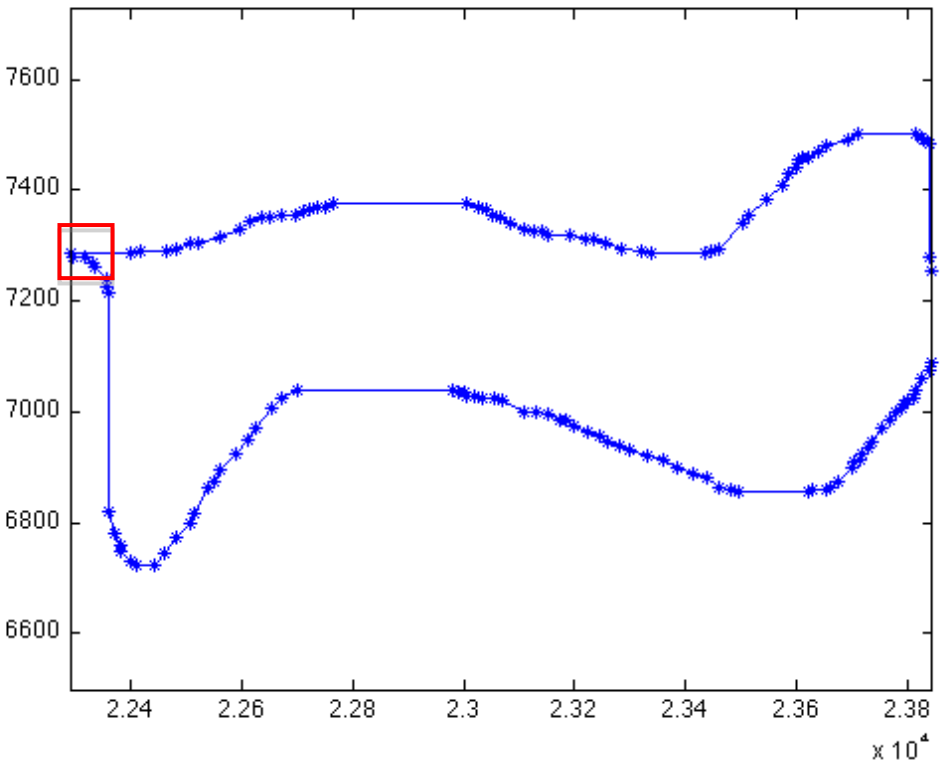

x 10^4
4.6
4.58
4.56
4.54
4.52
4.5
4.48
4.46
4.44
4.42
2.75
2.8
2.85
2.9
2.95
x 10^4

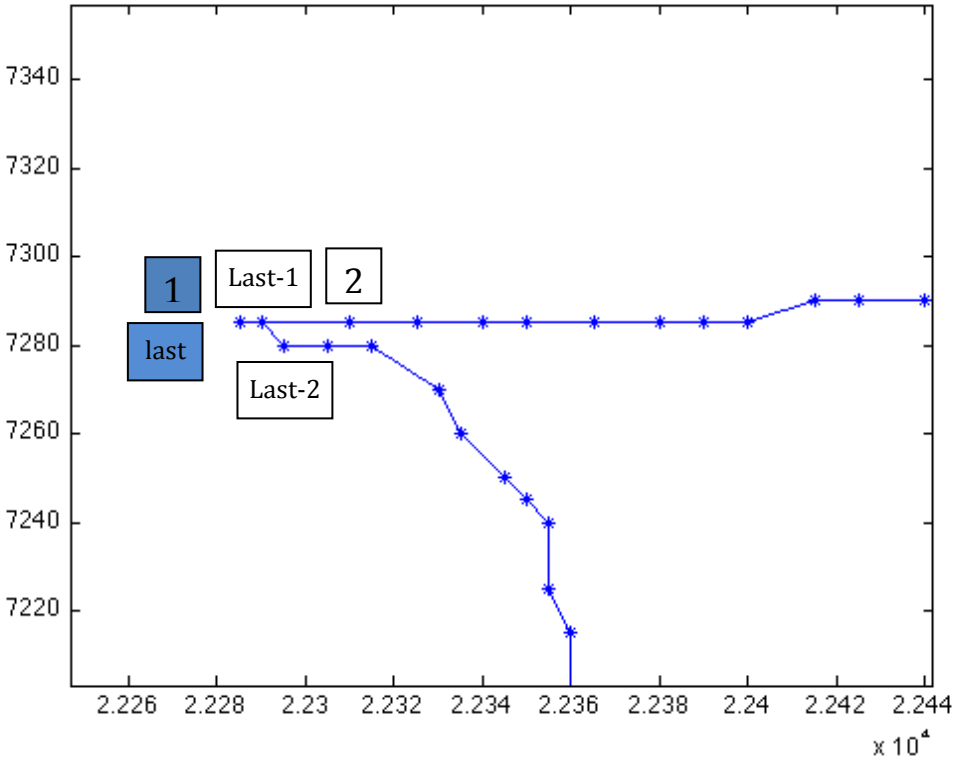

x 10^4
4.406
4.405
4.404
4.403
4.402
4.401
2.752
2.753
2.754
2.755
2.756
2.757
2.758
2.759
x 10^4
Last-2
2
1
last
Last-1

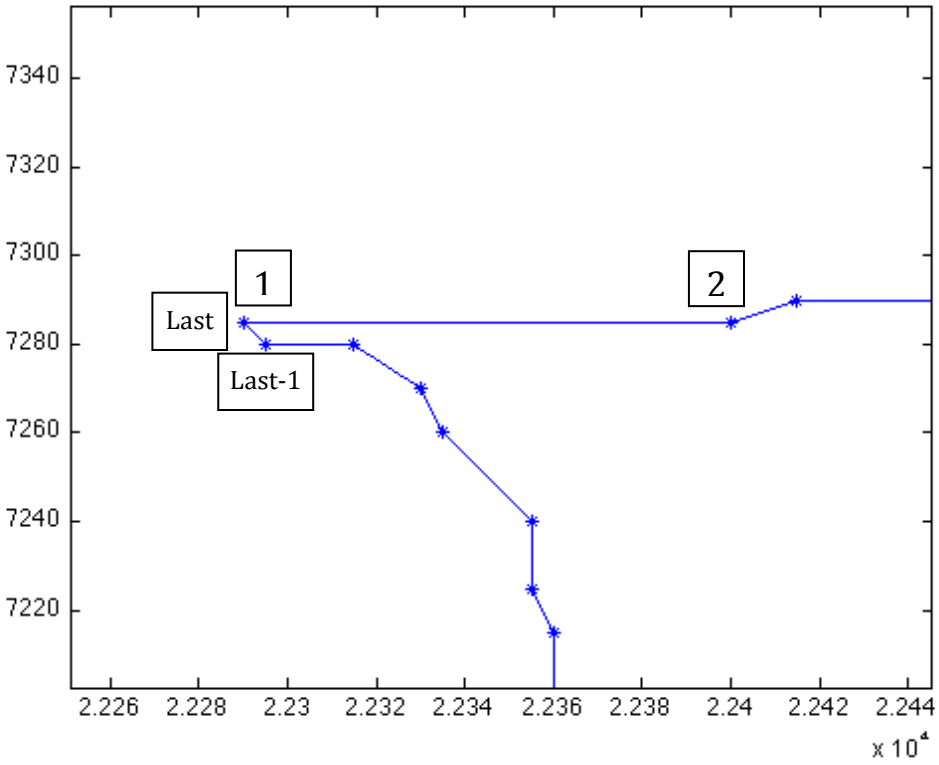

x 10^4
4.406
4.405
4.404
4.403
4.402
4.401
2.752
2.753
2.754
2.755
2.756
2.757
2.758
2.759
x 10^4
Last-1
Last
1
2

Case 3:

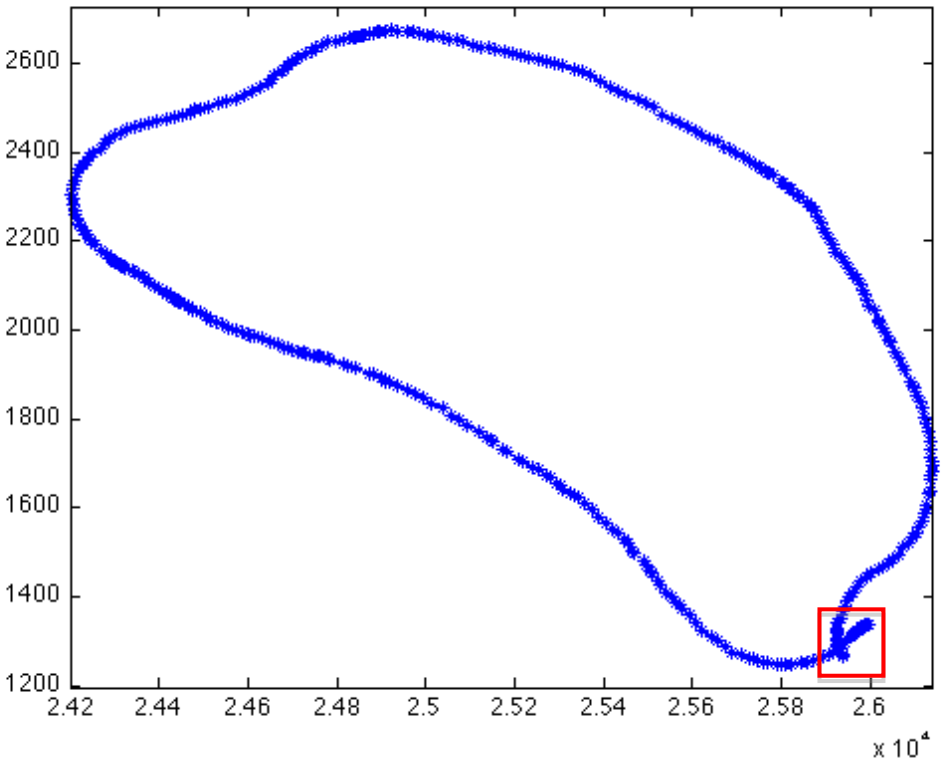

7600
7400
7200
7000
6800
6600
2.24
2.26
2.28
2.3
2.32
2.34
2.36
2.38
x 10^4

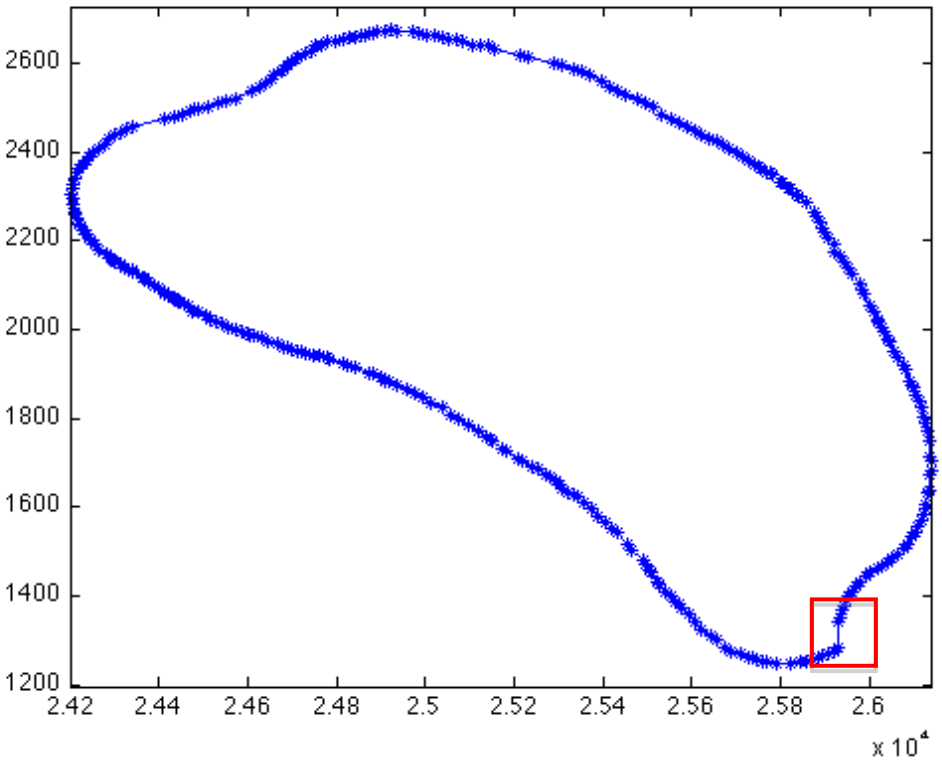

7600
7400
7200
7000
6800
6600
2.24
2.26
2.28
2.3
2.32
2.34
2.36
2.38
x 10^4

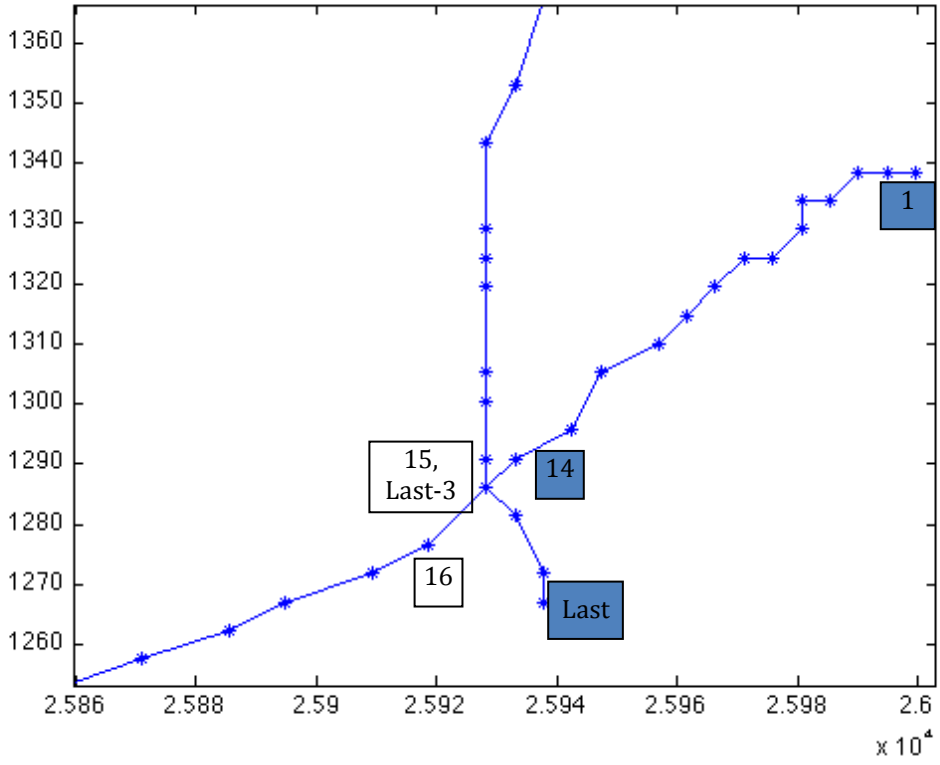

7340
7320
7300
7280
7260
7240
7220
1
Last-1
2
last
Last-2
2.226
2.228
2.23
2.232
2.234
2.236
2.238
2.24
2.242
2.244
x 10^4

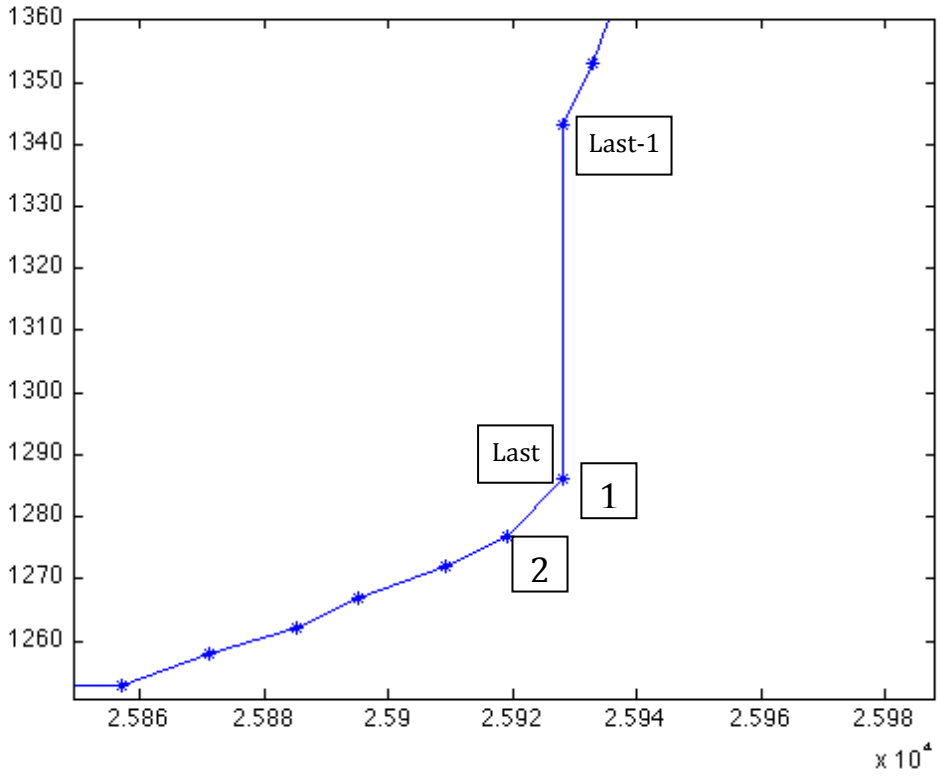

7340
7320
7300
7280
7260
7240
7220
1
2
Last
Last-1
2.226
2.228
2.23
2.232
2.234
2.236
2.238
2.24
2.242
2.244
x 10^4

Case 4:

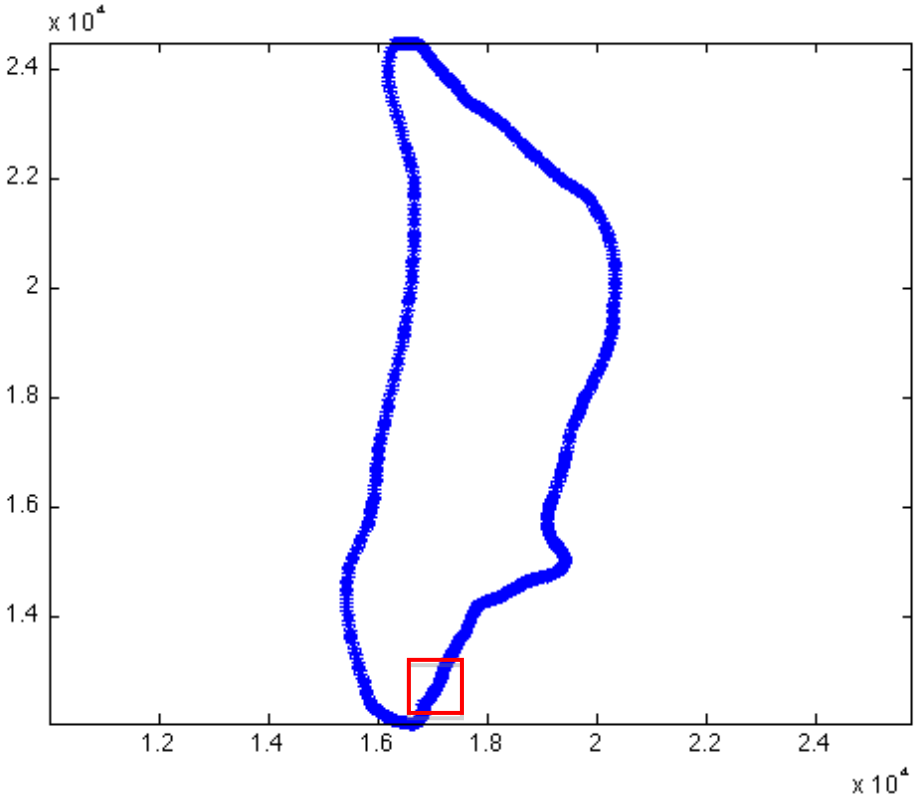

2600
2400
2200
2000
1800
1600
1400
1200
2.42
2.44
2.46
2.48
2.5
2.52
2.54
2.56
2.58
2.6
x 10^4

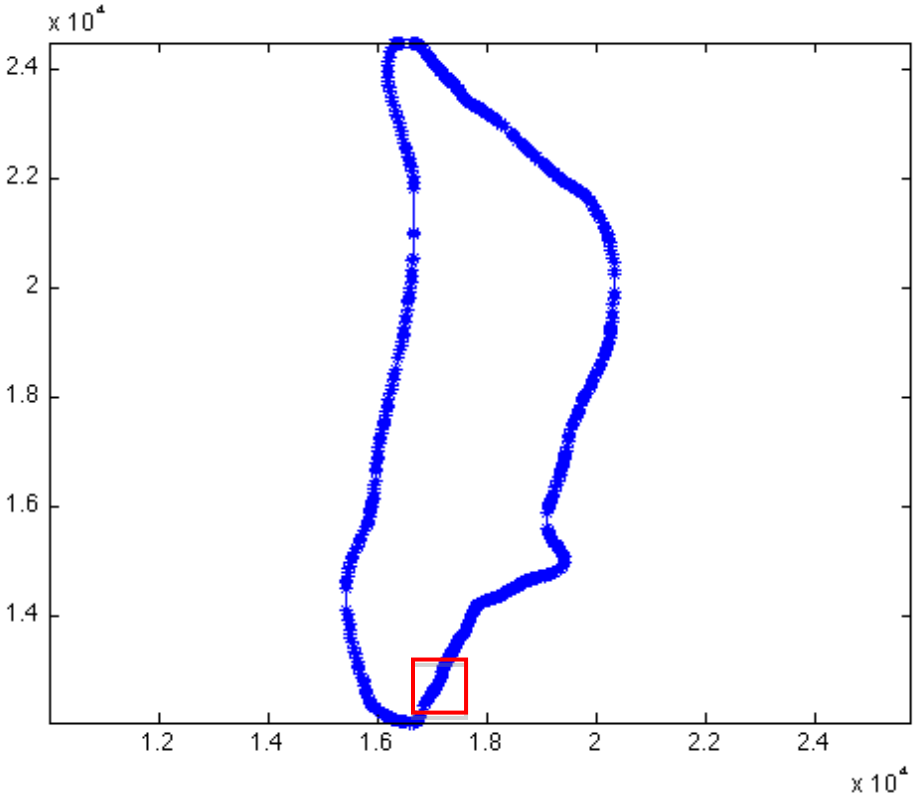

2600
2400
2200
2000
1800
1600
1400
1200
2.42
2.44
2.46
2.48
2.5
2.52
2.54
2.56
2.58
2.6
x 10^4

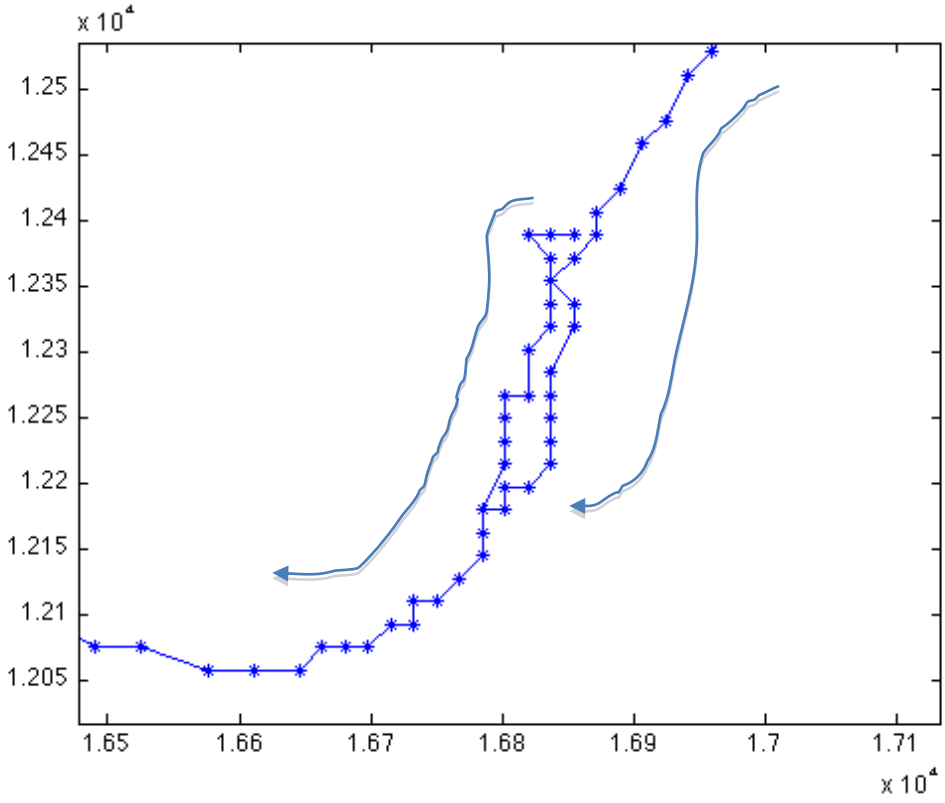

1360
1350
1340
1330
1320
1310
1300
1290
1280
1270
1260
2.586
2.588
2.59
2.592
2.594
2.596
2.598
2.6
x 10^4
1
15,
Last-3
14
16
Last

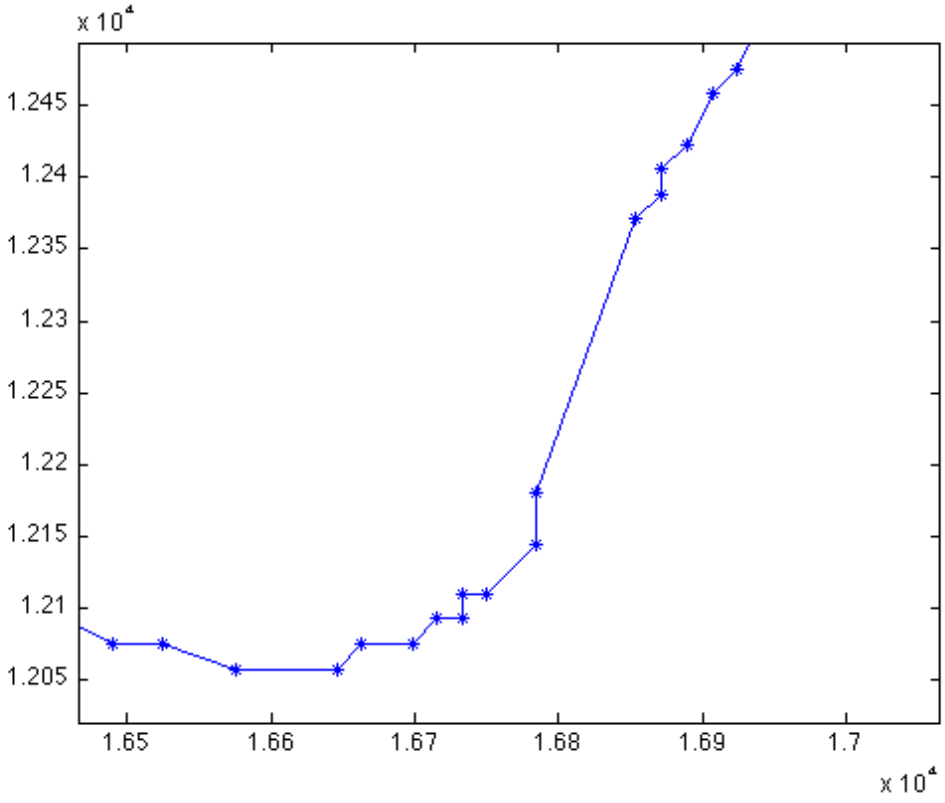

1360
1350
1340
1330
1320
1310
1300
1290
1280
1270
1260
2.586
2.588
2.59
2.592
2.594
2.596
2.598
x 10^4
Last-1
Last
1
2

Case 5:

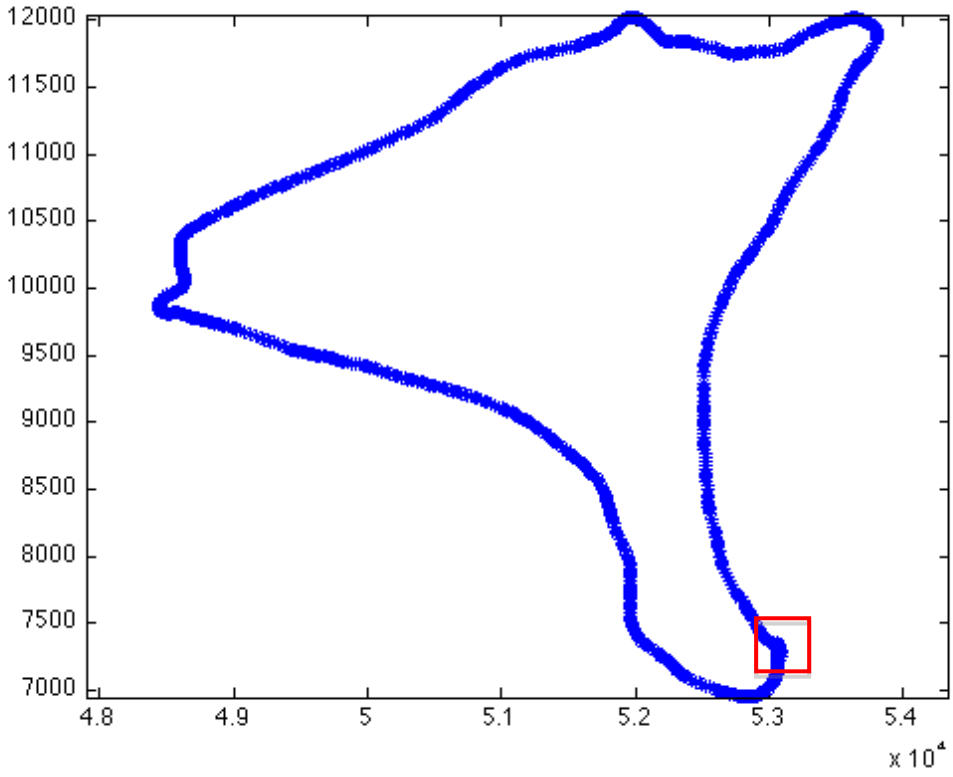

x 10^4
2.4
2.2
2
1.8
1.6
1.4
1.2
1.4
1.6
1.8
2
2.2
2.4
x 10^4

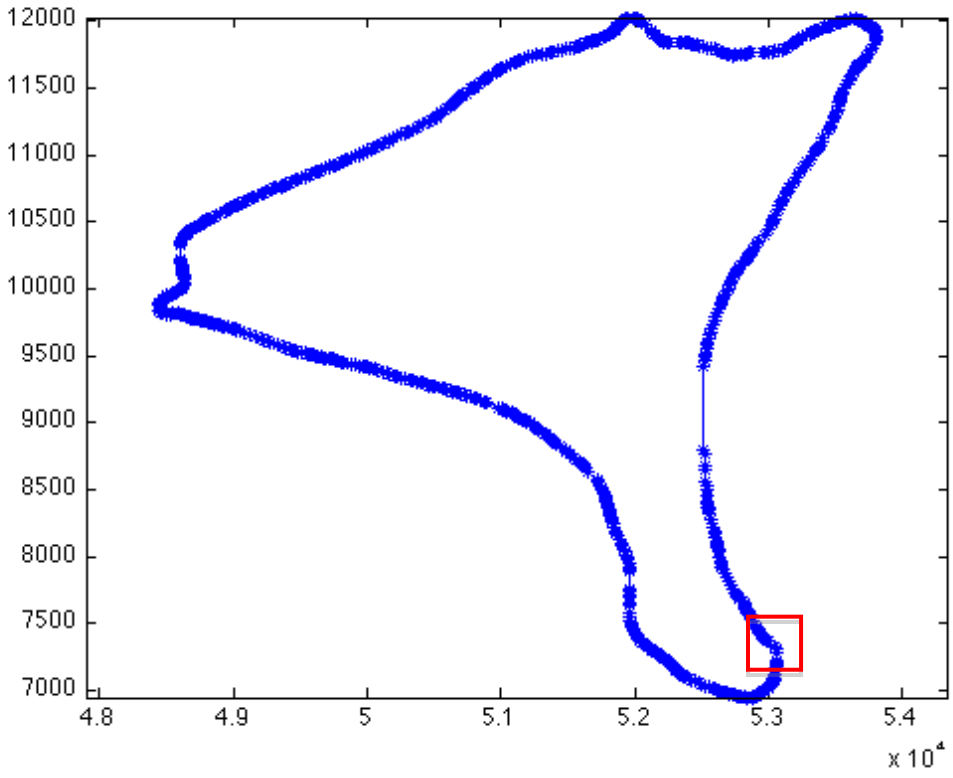

x 10^4
2.4
2.2
2
1.8
1.6
1.4
1.2
1.4
1.6
1.8
2
2.2
2.4
x 10^4

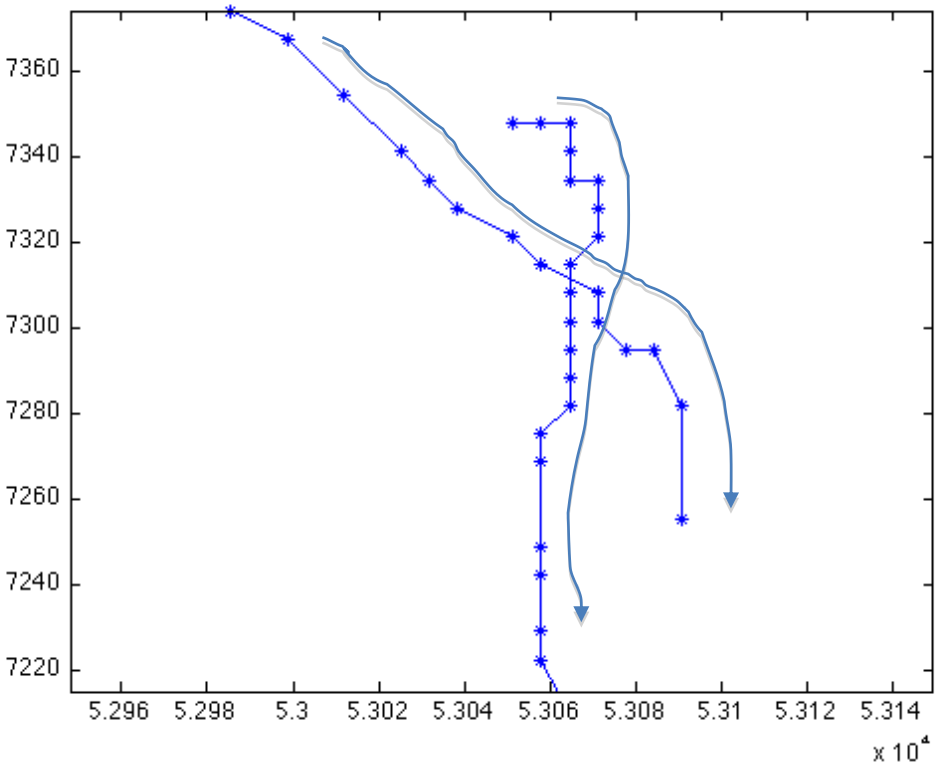

x 10^4
1.25
1.245
1.24
1.235
1.23
1.225
1.22
1.215
1.21
1.205
1.65
1.66
1.67
1.68
1.69
1.7
1.71
x 10^4

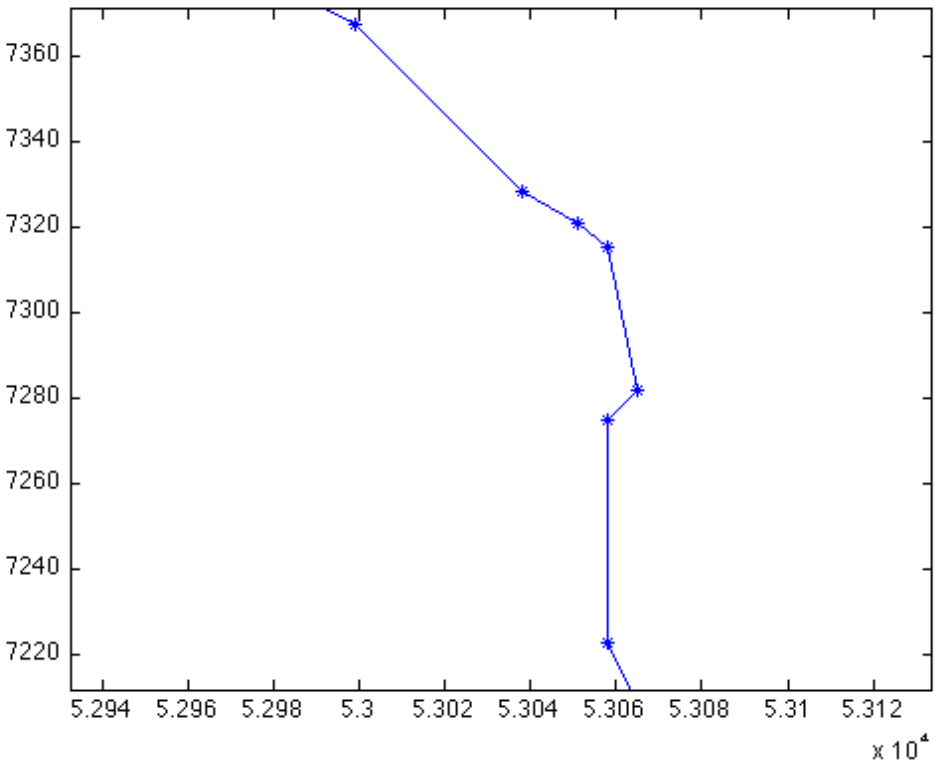

x 10^4
1.245
1.24
1.235
1.23
1.225
1.22
1.215
1.21
1.205
1.65
1.66
1.67
1.68
1.69
1.7
x 10^4

Case 6:


12000
11500
11000
10500
10000
9500
9000
8500
8000
7500
7000
4.8
4.9
5
5.1
5.2
5.3
5.4
x 10^4


12000
11500
11000
10500
10000
9500
9000
8500
8000
7500
7000
4.8
4.9
5
5.1
5.2
5.3
5.4
x 10^4


7360
7340
7320
7300
7280
7260
7240
7220
5.296
5.298
5.3
5.302
5.304
5.306
5.308
5.31
5.312
5.314
x 10^4


7360
7340
7320
7300
7280
7260
7240
7220
5.294
5.296
5.298
5.3
5.302
5.304
5.306
5.308
5.31
5.312
x 10^4