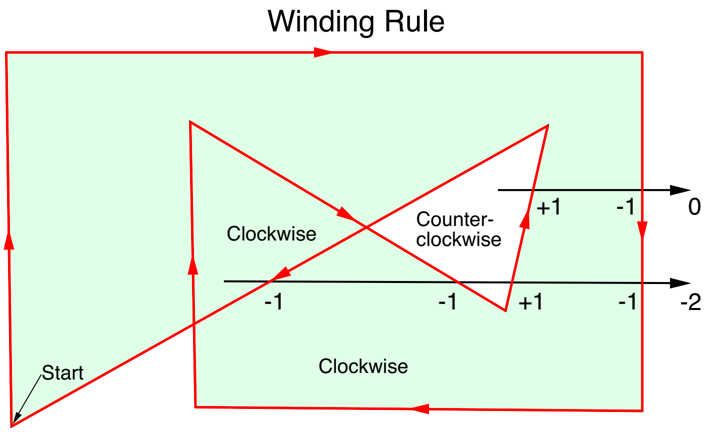
For a demonstration of segmented Bezier curves, execute this in Igor:

```
DisplayHelpTopic "Segmented Bezier Curves"
```

Polygon and Bezier Curve Fill Rules

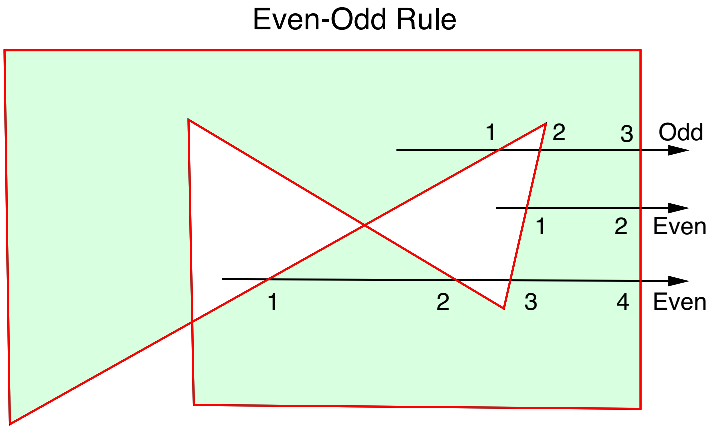
Simple polygons and Bezier curves with no intersecting edges are filled unambiguously - points within the shape are filled with color. But how do you decide what is inside and what is outside if the shape has intersecting edges?

There are two rules in common use: the Even-Odd rule and the Non-Zero Winding rule. Igor uses the Winding rule by default:



Starting at a given point, draw a line to infinity. If an edge is crossed and that edge is drawn from up to down (or clockwise with respect to the start of the line) subtract one. If the edge is drawn from down to up (or counterclockwise), add one. If the result is non-zero, it is inside and should be colored.

You can request the Even-Odd rule using `SetDrawEnv fillRule=1`. The fillRule keyword applies only to polygons and Bezier curves created with **DrawPoly** and **DrawBezier**, not to those created manually. Manually-created polygons and Bezier curves follow the winding rule by default; you can change the rule using the Modify Polygon or Modify Bezier dialogs.



Starting at a given point, draw a line to infinity. Count the number of edges crossed. If the result is an odd number, the point is inside. If it is even, it is outside.