

Hi again Choy,

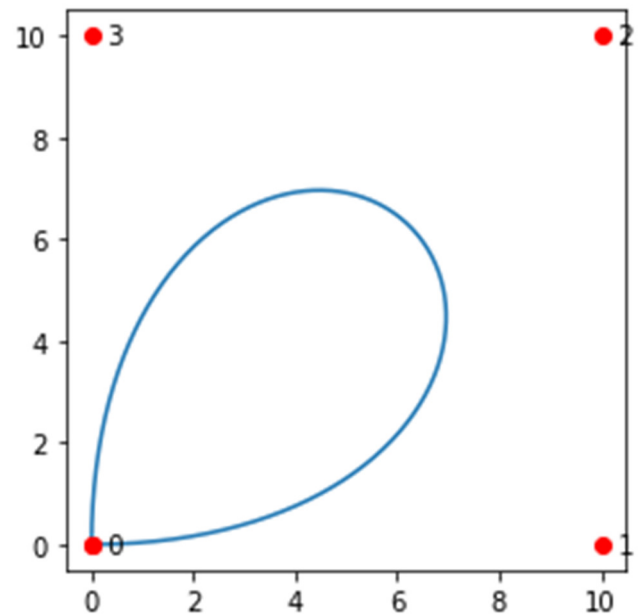
Having looked deeper into this, I have been reminded that Bezier curves form curves based on data provided as “control points”. From the basics I can see about Bezier curves and their GML implementation, it looks like Bezier curves rarely actually hit those control points (beyond the first and last point), but instead are confined within a “convex hull” defined by the control points. This is very different to the current implementation using Splines for closed objects, as it means that the actual curves defining the limits are within this convex hull, rather than within curves that pass through each of these points, as is currently the case.

The attached plot shows an example, for clarity. I defined 5 points to create a square (duplicate points at the origin to close shape). I fed these into a Bezier function in Python. It created a 4th order Bezier curve. This curve does not come near three of the control points. If the same points were used for SigWx Object generation, it would create a shape that was approximately circular, passing through all of the points.

This might be fine for jet objects, but for all other closed object types this would lead to a marked shrinking of all hazard areas. Perhaps this is something I have misunderstood, but would need clarification.

Many thanks,

Graeme



Good morning Choy,

I apologise for taking a while to get back to you on this, I was working in the Met Office College delivering training when you sent the email, and although I have been back working in Aviation for the past week or so, I have had other priorities, and so it has taken me a while to get to your email.

I've read through the whole discussion below, and it is quite interesting, although it does mean I will need to go back and study notes I made on how splines are described, and a new set of notes on Bezier curves! I certainly lack the knowledge in this level of geometry, and will struggle to determine the differences between the different types of curve.

The main argument I picked out, however, is that if we use Bezier curves, these will be consistent and non-ambiguous between software providers in terms of how the curves are plotted. It sounds like there are multiple ways to interpret the term spline, meaning different users may take the same points and plot different curves. This sounds like an advantage to using Bezier curves.

I think the way manual SigWx plots have been created so far by Operational Meteorologists has simply been “the software draws a curve for me”, with no comprehension of the underlying process of the various ways such a curved line could be drawn. The Op Mets have the advantage of being able to see the output in

their software and drag points around, such that they can make the curve fit the data appropriately no matter what type of curve is drawn.

With the Python-generated SigWx Objects, I think I would just need to play with plotting Bezier and spline curves through sample points of data, and consider whether the differences are significant to push a decision one way or the other. I suspect the differences will not be significant enough for me to be able to determine on as “better” or “worse” than the other, however.

So in summary, let me go away and have a play with Bezier and spline curves and get back to you. My uninformed opinion based on the feedback from Boris and Jan is that Bezier curves would be more consistent, making this a better option, but I really am lacking sufficient knowledge to make that call with certainty.

Many thanks,

Graeme