FSComp software current state and future

September 30, 2019

This is a review of current state FSComp software, issues found in source code, which may cause problems during competition scoring.

1 Turnpoint crossings.

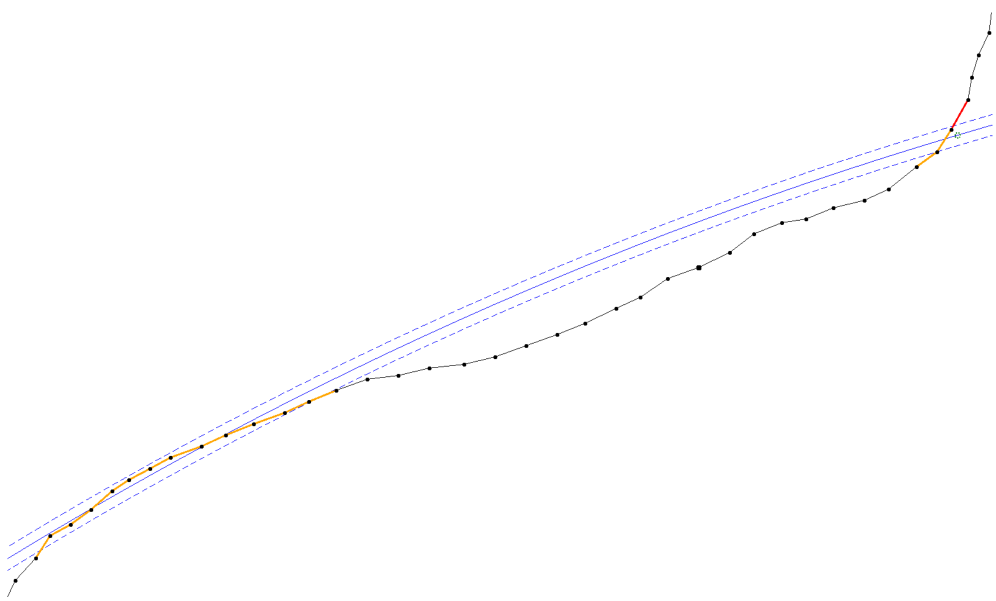

Figure 1: Tracklog and section of a cylinder boundary. Potential cylinder crossings are displayed in orange, and the red one is used for scoring. Solid blue line is a cylinder boundary, and dashed lines pair define tolerance band.

Currently in FSComp (R1.2, last commit 438aacb) turnpoint crossing is checked in the following way. Suppose given cylinder turnpoint with centered at $C = (\phi_C, \lambda_C)$ and radius R, and tolerance τ is set (normally for Cat. 1s it's $\tau = 0.1\% = 0.001$), which together define a pair of cylinders with radiuses $R_G = R + \max(R \cdot \tau, 5.0)$ and $R_L = R - \max(R \cdot \tau, 5.0)$ – a tolerance band around original cylinder (see Fig. 1). Potential crossing is a consecutive pair of tracklog points, where either one or both points lie within the tolerance band, or both lie outside of the band, but on different sides of it (see Fig. 2).

For scoring, in case of ESS cylinder and all ordinary cylinders, first (timewise) crossing is used (which is ok), but for SSS the last one is taken (among those, which will provide maximum amount of next TPs crossed). That probably was done for Elapsed Time or multiple start gates type of tasks, where indeed it's best for pilot to take latest start time in order to get smallest task time and maximize time points. But for normal, most often used, Race to Goal type of task this potentially may cause problems with leading points in some cases. Consider configuration as in Fig. 1 and assume it's SSS cylinder and pilot flies ENE towards next turnpoint. In the way currently leading points are calculated, he'll start scoring leading points at the "second" exit from the cylinder, way later than should, and as it's the beginning of task, will get noticeably less leading points.

It's interesting to notice, that direction of crossing is stored, but not checked, except of one case when pilot doesn't make it to the next turn points after SSS, and that turnpoint is inside SSS cylinder, but not concentric.

FSComp erroneously treats it as Exit type, and thus, unless pilot, approaching it from the outside, when start is open, enters and then makes exit and then enter again maneuver, he won't be scored neither start, nor the rest of the distance. This check doesn't make much sense (as any other potential Enter/Exit check - this concept is harmful and brings only confusion and useless discussions and arguments at takeoff) and probably should be removed as causes nothing but problems (see [1] for example of such problem).

For all turnpoints except the last one (goal) proper distance on WGS84 ellipoid calculation is used (Andoyer method). Further though, to calculate coordinates of cylinder crossing, UTM projection is used, which causes noticeable errors, but those used only for visualization, not computation in scoring (see green dashed circle in right top corner of Fig.1: it should be a point on tracklog, where it crosses the cylinder). Goal line crossing check and semicylinder methods use both Andoyer and UTM based computations and under certain cicumstances may be imprecise. As normally goal line is short and potentially large distances there are calculated with Andoyer method, errors should not be significant, but requires further investigation. In general it might be worth to remove all UTM based calculations that might affect scoring in any way.

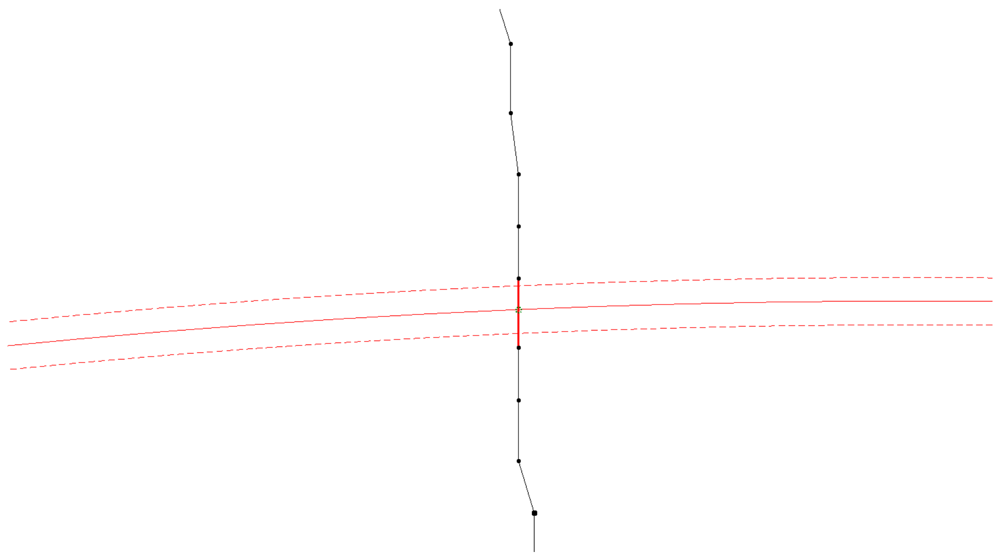

Figure 2: Tracklog and section of a small radius ESS cylinder boundary. There is only one crossing, with points lying on different sides of the tolerance band.

The way time of crossing is calculated is implemented in unnecessary complicated and it depends on whether crossing happened over cylinder itself or one of tolerance defined band cylinders. In some cases time of crossing is interpolated between endpoint times. This looks like is coming from times when instruments usually stored track points in intervals significantly larger than 1 second. This approach in general doesn't look correct, as one may produce example when it's beneficial to remove some track points around ESS crossing moment to get better time results from interpolation (such as when flying into ESS through strong wind gradient). It may be worth to simplify it and just use time of first turnpoint inside ESS cylinder (maybe minus one second if it's desired be consistent with current calculations). This will encourage usage of 1 second intervals in tracklogs and will be more in spirit how it's done when leading points are calculated (and some part of track log is missing) – use information that is certain and do not interpolate.

2 Distance calculations and task visualization.

Except for checking turnpoints crossing case, where proper distance computation on WGS84 ellipsoid is used, UTM projection with zone defined by start point is common across the codebase. Task distance, distance flown, remaining distance to ESS used in leading points calculation, task visualization – all are tainted by usage of UTM projection. There are few issues with UTM usage.

First, there is discontinuity at equator: tasks set with cylinders crossing equator line may be incorrectly evaluated (Equador, Kenya, Indonesia and few other countries may be affected). UTM was designed to have only positive values in coordinates of projection, thus for South hemisphere False Northing of 10000km is added, which causes wrong results in distance calculation (see Fig. 3).

Second, depending where exactly first turnpoint of task w.r.t. UTM zone is, local task scale can differ significantly. This means, for example, if takeoff is close to zone center, all cylinders will be calculated 0.04% bigger than defined on WGS84 ellipsoid [1] (in the current codebase cylinders boundary points are sampled using UTM, but distances are further calculated with Andoyer), on the edge of the zone it will be up to roughly 0.097% smaller than defined. As task may start in one UTM zone and continue into another (but all lat-long coordinates are converted into original zone still), local scaling factor error may go well over 0.1%. Curiosly, this will not affect scoring in significant way, as cylinders crossing are still checked with Andoyer algorithm (thus cylinder radiuses are used as defined, UTM is not used there), but leading points calculation and distance flown (for bomb outs) is slightly affected.

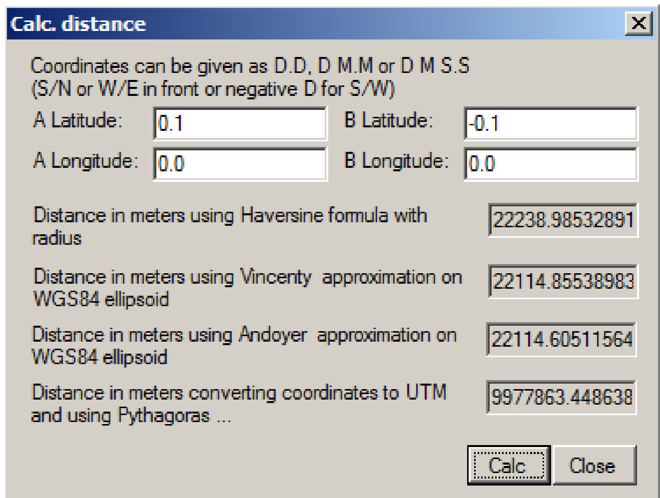

Figure 3: Different distance calculation algorithms implemented in FSComp for two points on different sides of Equator. All algorithms more or less agree except UTM based.

More importantly, for visualization UTM produced cylinders are shown, and this may cause the following problem. Suppose pilot didn't make a turnpoint and comes to the scorer to see what happened (see Fig. 4). The turnpoint radius is 30km and tolerance is 0.1%, and thus 30m. FSComp shows though, that pilot flew as close as 27m from turnpoint. Should this turnpoint be scored? If scorer doesn't know how it's implemented internally (and there is nobody around sufficiently knowledgeable), there is a chance that scorer will decide to hack around (like raise tolerance for given task) and make FSComp to score this turnpoint. This would be a wrong decision, as FSComp calculated everything in a correct way, but visualization is wrong. In Fig. 5 it's clearly seen that neither of tracklog points fall into tolerance band of the true cylinder line.

Thus UTM usage should be abandoned to the maximum extent in future. Though generic Transverse Mercator projection, adapted for PG/HG tasks may have some use for scoring. Its precision is the subject of further investigation, but it appears for small area tasks should be good enough. It may well be that another projection with more directionally uniform scaling factor change could be used instead. In any case, from standpoint of precision, it's still best to use proper WGS84 ellipsoid computation, unless performance aspect becomes really important. For laptop computations it doesn't appear to be an issue.

[1] This is due to 0.9996 global scaling factor used in UTM

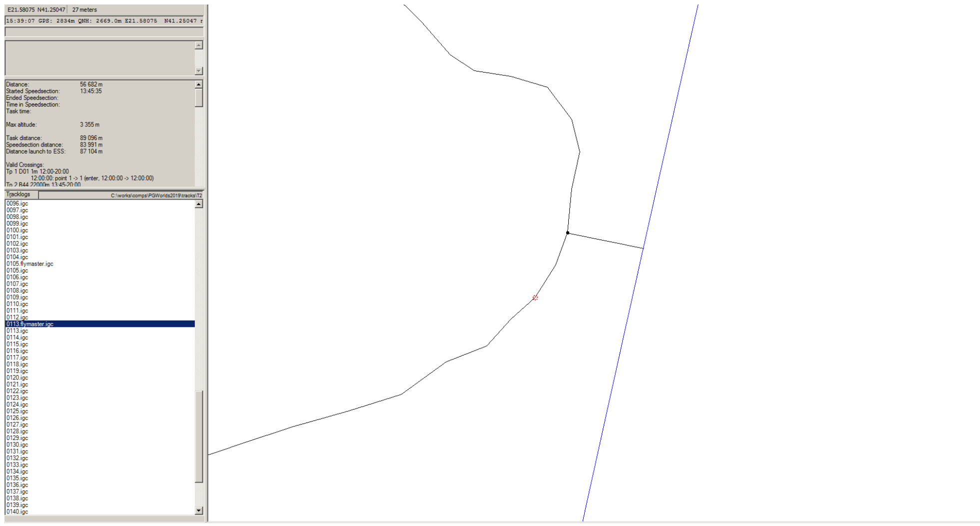

Figure 4: Pilot didn't reach turnpoint of 30km radius. Tolerance is set to 0.1%. FSComp would show distance 27m, which is less than tolerance 30m. Turnpoint should be scored?

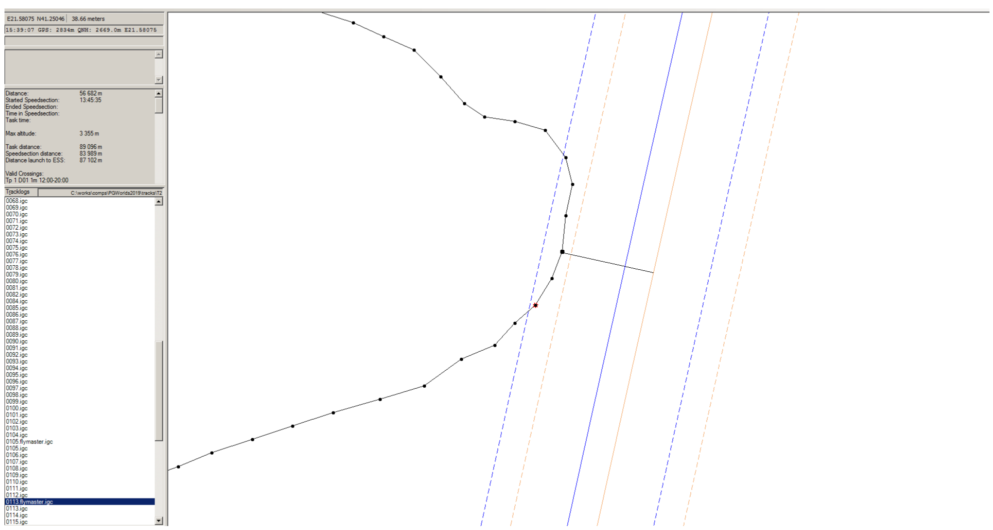

Figure 5: Situation from Fig. 4, but with added cylinder edge, calculated using proper WGS84 model (direct Vincenty method) and tolerance bands in dashed lines for both true cylinder edge (in orange) and estimated with UTM (in blue)

3 Airspace violation checks and visualization.

There are few issues in the way airspace violations are checked:

1. Cylindical airspaces (those which are defined by central coordinate and radius in nautical miles) are first converted into polygonal type by sampling the circle. Sampling is done in UTM coordinate space, which, given huge radiuses in some cases, may cause noticeable errors.

2. Polygonal airspaces are checked by proper polygon point inclusion algorithm in (ϕ, λ) space - this is equivalent of using equirectangular (Plate Carree) projection. In case of large distances between points, defining a polygon edge, straight line in the projection may differ significantly from geodesic line, which may cause incorrect check for points close to the airspace (see [2] for more details). It may be worth to sample such lines along geodesic to ensure sufficient precision.

3. In the code currently it appears that barometric (QNE) altitude is also always checked (in addition to whatever is set), no matter what scoring altitude type is set. On low pressure day it may cause reports of pilots in airspace, even though GPS scoring altitude was set and everybody is clear according to GPS. Not sure that's desired.

4. When tracklog is visualized, segments before and after point in violation are not drawn, which may cause confusion.

5. In the code airspace checks are not part of scoring, but part of visualization. It may be worth to ensure that airspaces are part of task (and competition if necessary) definition and should be checked and scored when scoring is done, so it will not be occasionally forgotten. Also penalties should be automatically assigned (as it looks to be time consuming pain now).

4 Leading points calculation

Speedsection distance is defined Section 7F as "the path of shortest distance from start of speed section to end of speed section that touches all turnpoint cylinders". Currently it's computed as length of subpath between SSS and ESS cylinders of optimal full task route (path between takeoff cylinder and goal cylinder), which is not correct and differs significantly (see Fig. 6). Correct SS distance computation is essential for leading points computation and difference can be way more than 10pts (on a task with around 150.0pts for leading are available).

During leading coefficient computation, first member of time graph is ommited. This may be due to the fact that ESS distance is computed incorrectly, thus once pilot crosses SSS, it may be computed that pilot had flown alread few kilometers in SS, and once things get squared, that addition is huge, and makes ridiculous changes in LC. With proper SS distance computation it's not possible and first member can be used (and it does make difference, as expected in the beginning of the task).

For some reasons power 0.666 is used instead of $\frac{2}{3}$. This makes small changes (like 0.1pts for leading points score), but due to rounding it can easily change resulting score and ranking. There is no need for that approximation.

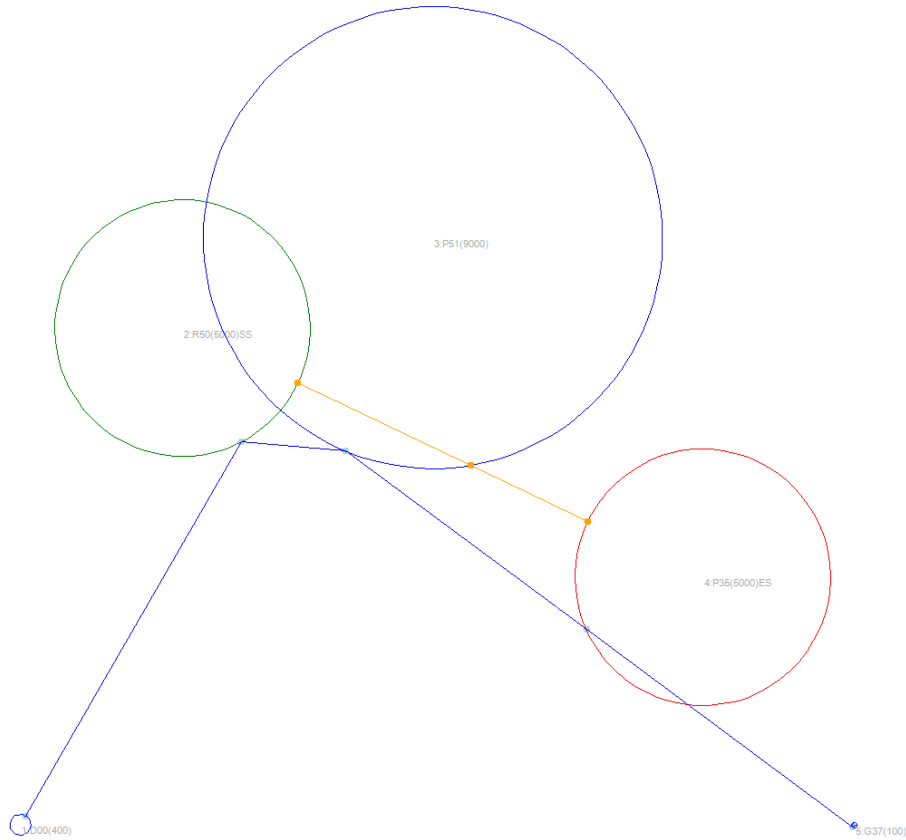

Figure 6: Optimal task route (blue) and optimal speed section route (in orange).

5 Miscelanous

Task cylinders are drawn as cardinal spline curves, interpolating between sampled points on cylinder starting with eastmost point. This causes barely visible corner there and slightly incorrect general shape. Method for closed curves should be used instead. Also for turnpoint boundary points sampling integer radius is used, which together cause small artifacts in visualization (see Fig. 7).

Maximum turnpoint radius is set to be 100km. There was a precedent of setting 102km cylinder at PWC Portugal in 2019 (task was cancelled due to weather though). Once UTM computation are not used any more, it would be safe to define and use way bigger cylinders.

If someone was wondering, what's projection used for visualization of task, tracklog, airpsaces, etc. - it's equirectangular (Plate Carree) with aspect ratio forced to be 1.0 in the center of screen (not UTM, Web Mercator or whatever).

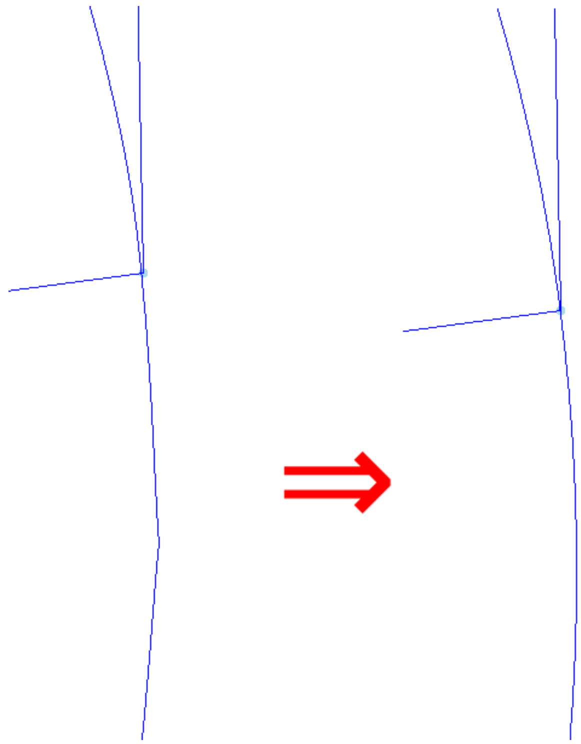

Figure 7: Artifacts in cylinder visualization.

References

[1] Dimov D. *FS and "start direction"*,
`http://www.paraglidingforum.com/viewtopic.php?t=96904`

[2] Neeley P., Narkawizs A. *Map Projection Induced Variations in Locations of Polygon Geofence Edges*
`https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170011116.pdf`
