# Simplifier.py, a basic python script for simplifying polygons

*Source files are available in* [*https://github.com/OrdnanceSurvey/os-simples.git*](https://github.com/OrdnanceSurvey/os-simples.git)

The python script Simplifier.py has been developed to read a geojson file, call a simplify method, and output the results to a new geojson file matching the format of the original. It has been written in python 2.7.18 and uses the shapely library (version 1.6.4).

The script takes 3 arguments – an input file name, a tolerance (in metres), and an output file name. The input file should reside in the same directory as the script (a fully qualified pathname could also be used, if preferred). The output file will always write to the working directory from where the script is run.

If the script is called with a single argument, it will use a default tolerance of ‘25’ and will write to the output file ‘output.geojson’. This is to allow users to drag and drop an input file onto the python script to save a bit of typing. An improvement would be to read default values from a configuration file.

If the script is called with a tolerance of ‘0’, it would generate a convex hull instead of calling the simplify method. *This is currently not implemented; as it turns out, results were less than favourable so it has been abandoned for now.*

Placeholder functions are currently included in the script to indicate how it may be extended. These include functions for validating the input, better handling of json properties, and for coordinate precision rounding (the existing method outputs coordinates to up to 15 decimal places, which is not very helpful in national grid coordinates).

The script operates by reading the geosjon geometry into a shapely geometry, and then simplifying it according to the parameter setting. The tolerance chosen will set a minimum distance between vertices and the *shapely.simplify* method smooths the polygon boundary by removing vertices that are ‘too close together’. The larger the number, the greater the effect of smoothing; for this exercise, I have used tolerances of 5, 10, 25, 50, 100, 200 and 500.

The simplify method keeps the general shape of the polygon, making it the ideal candidate for the required task and ensuring that the output continues to ‘best represent’ the coastline. File sizes can be reduced considerably with larger tolerances. The table below shows the results of each simplify process; the suffix on each of the file names represents the tolerance specified.

|  |  |  |
| --- | --- | --- |
| File | Number of vertices | File size |
| scottish\_mainland (original) | 502,159 | 12,622 KB |
| scotland\_5 | 133,780 | 3,037 KB |
| scotland\_10 | 76,950 | 1,750 KB |
| scotland\_25 | 34,875 | 797 KB |
| scotland\_50 | 17,496 | 402 KB |
| scotland\_100 | 8,360 | 194 KB |
| scotland\_200 | 4,061 | 95 KB |
| scotland\_300 | 2,631 | 62 KB |

# Screenshots (in QGIS with OS Open Zoomstack as background)

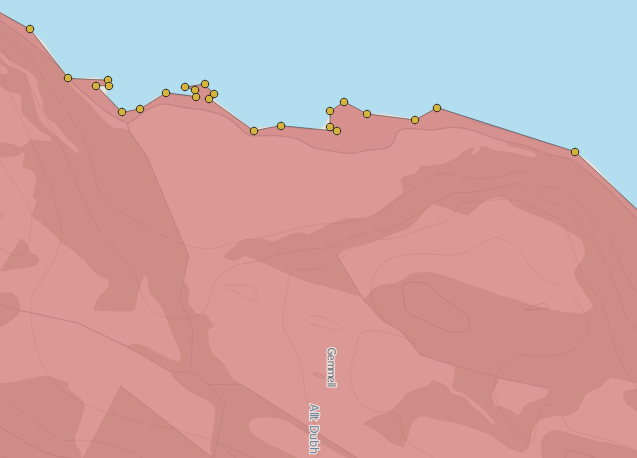
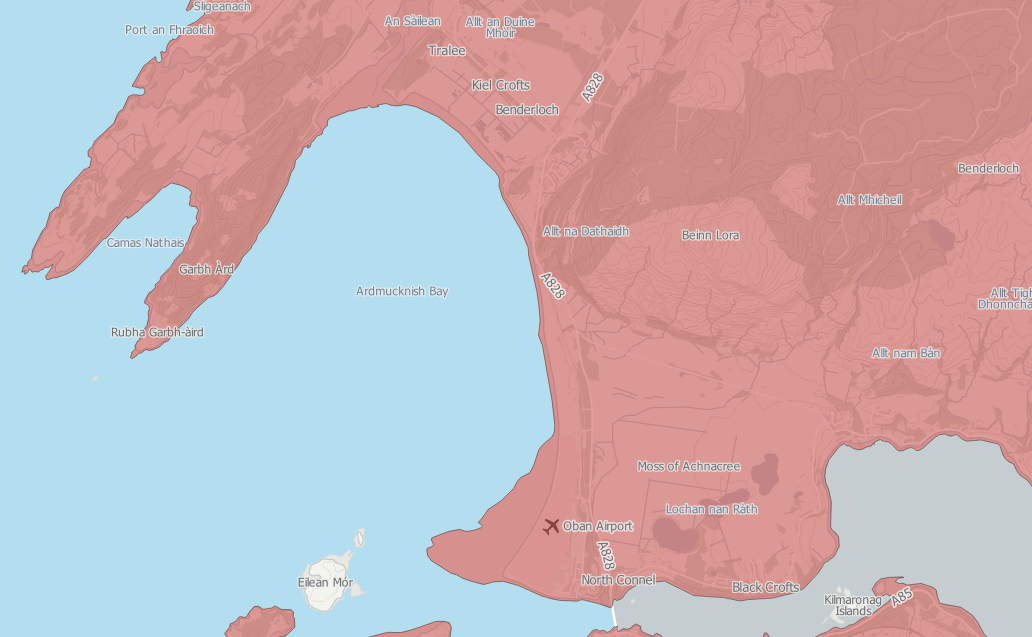
For each simplify process, I have included a screenshot of the full geometry, a close-up view that more clearly shows the effect of the simplification, and a screenshot of the vertices that remain.

The suitability of the output in each case will depend on the application but it seems that even a fairly generous tolerance of 25 metres produces a reasonable representation of the overall coastline. Not only that, it will output a file of just 797 KB, roughly 6% the size of the original file.

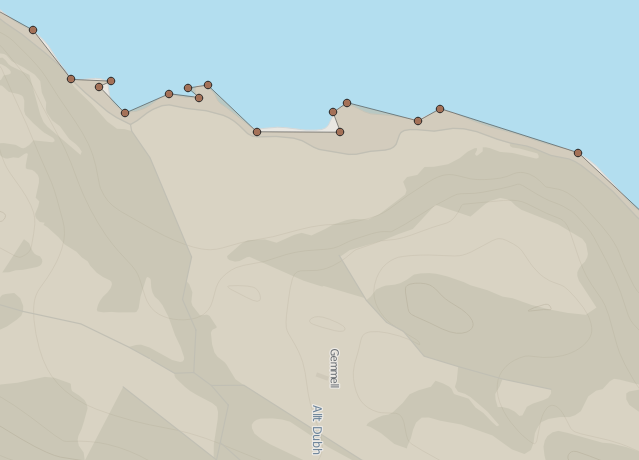
Original coastline geometry



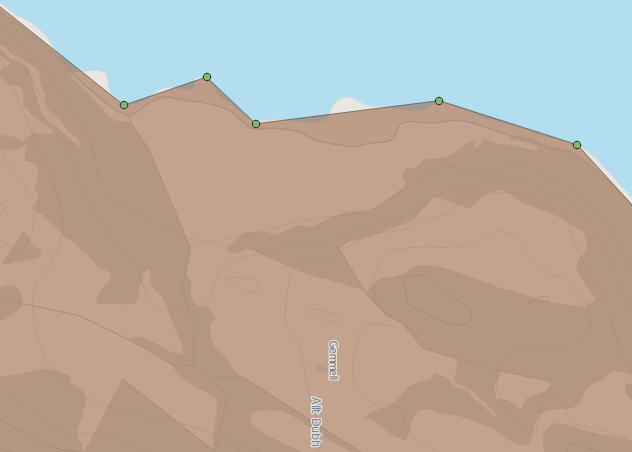
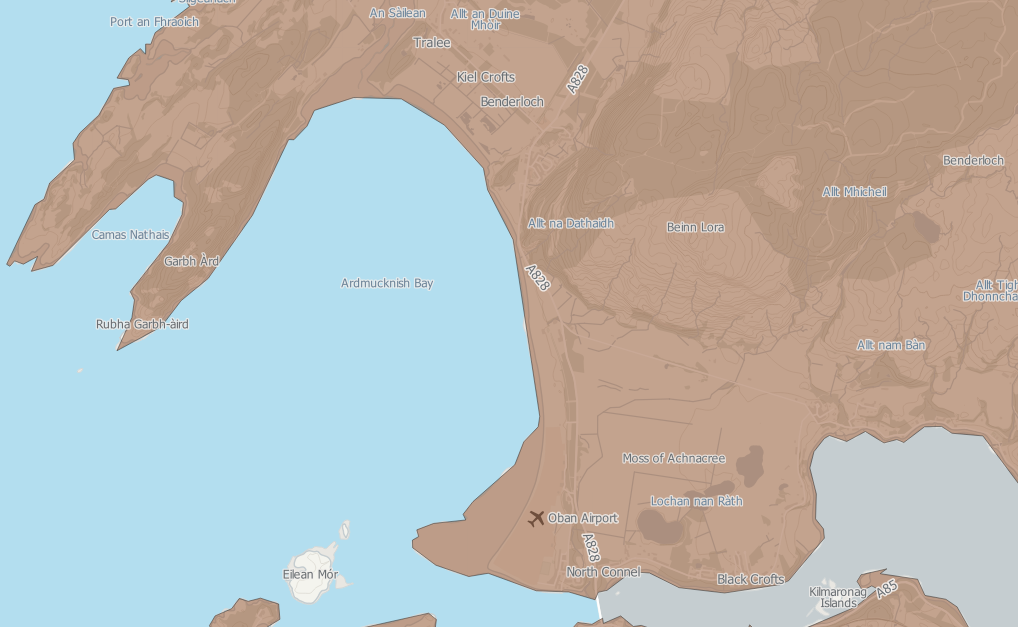
Generalised to 5 metres



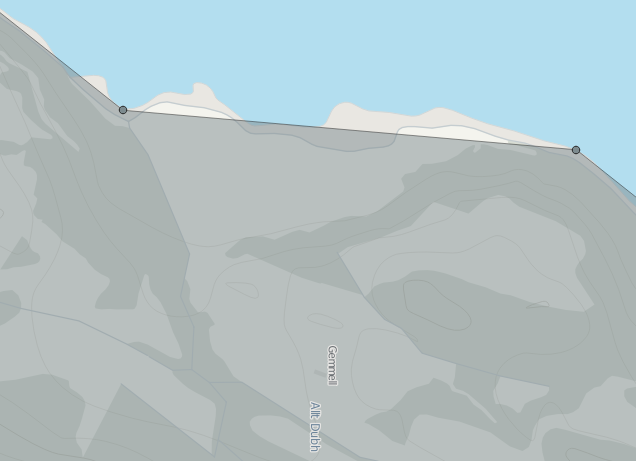
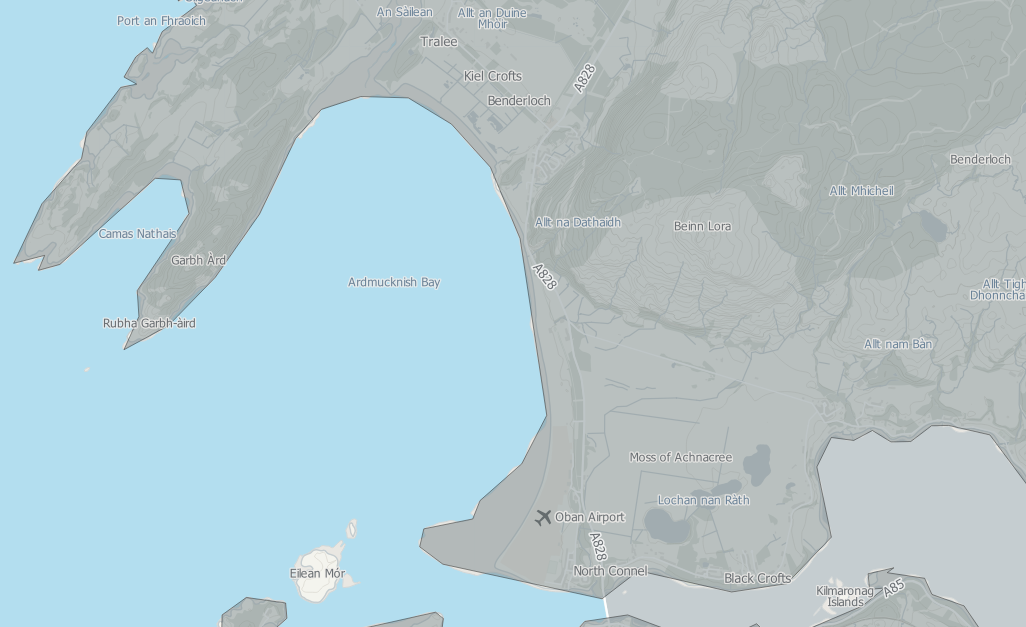
10 metre tolerance



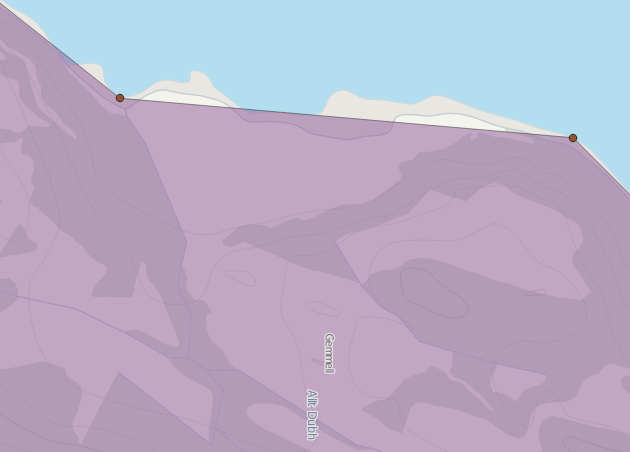
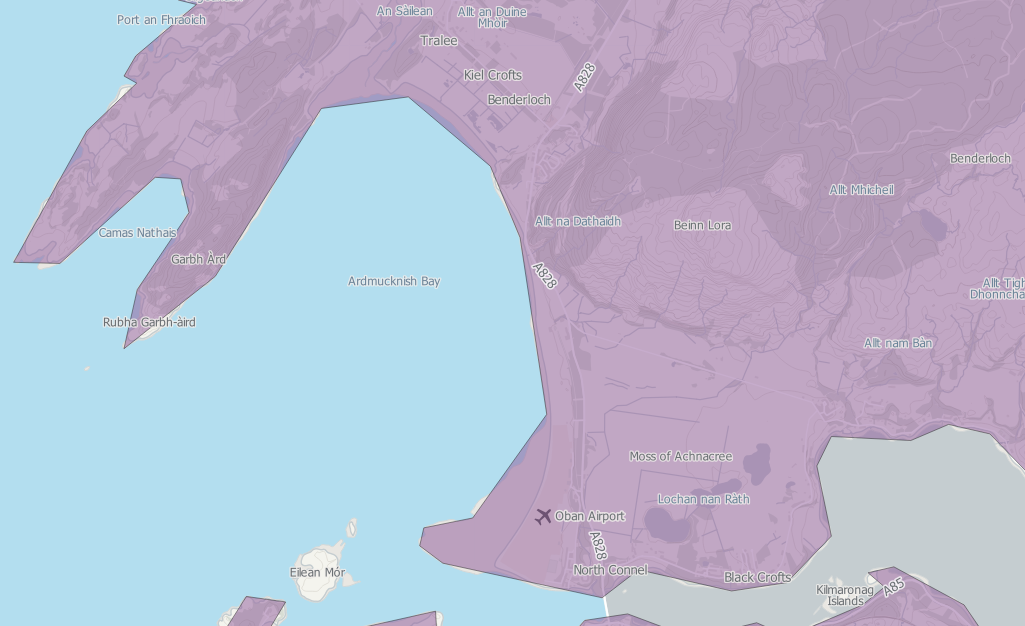
25 metres



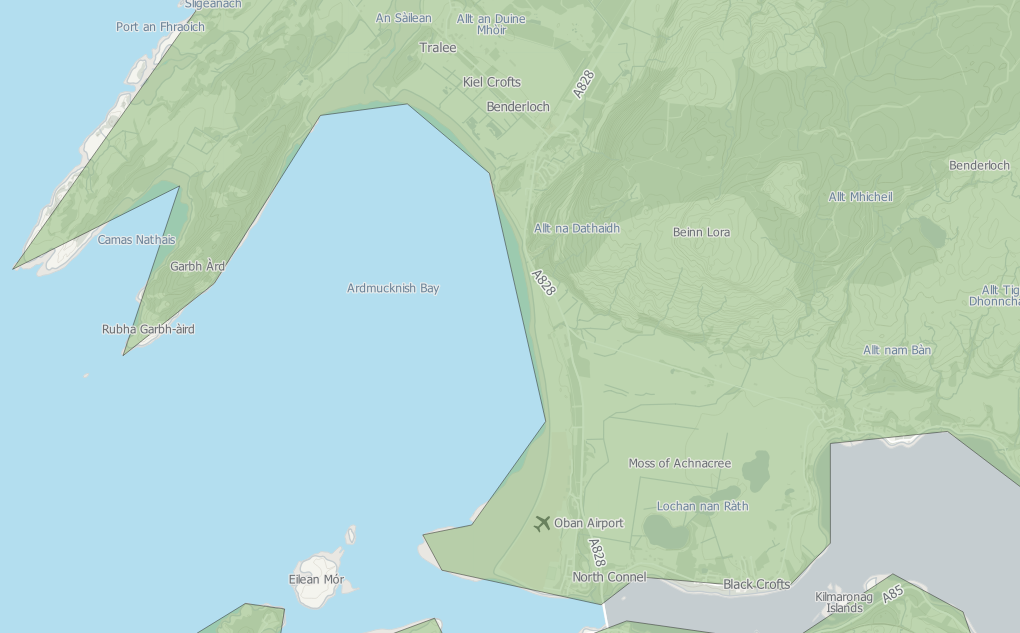
50 metres

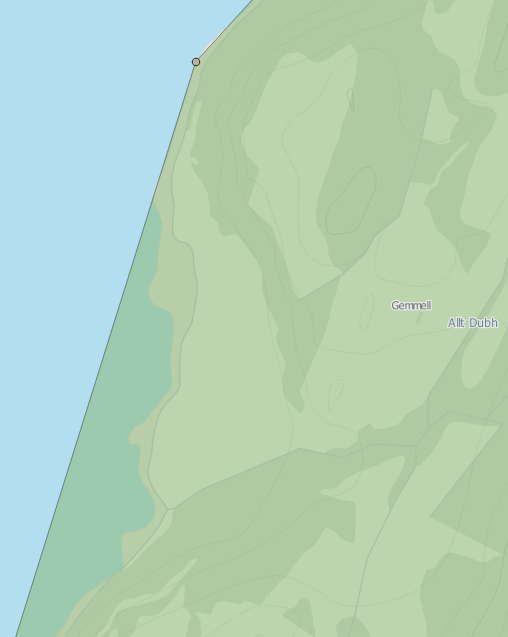


100 metres



200 metres





300 metres

