**Tropical cyclone wind field calculations** David N. Bresch, 20161226

A comparison between isimip (python), the implementation in MATLAB[[1]](#footnote-1) and the original climada wind field routine[[2]](#footnote-2).

Please note that the original isimip calculation has not been optimized for performance at all, while the original climada code has been pretty much optimized. Mainly two things lead to substantial speed gain, a) the use of a fast instead of proper haversine formula for distance on a sphere and b) using only the closest node to determine the wind speed.

Ad a) the faster distance calculations does merely correct with cos(latitude), which proves to be ok for the distances from the cyclone center that matter[[3]](#footnote-3):

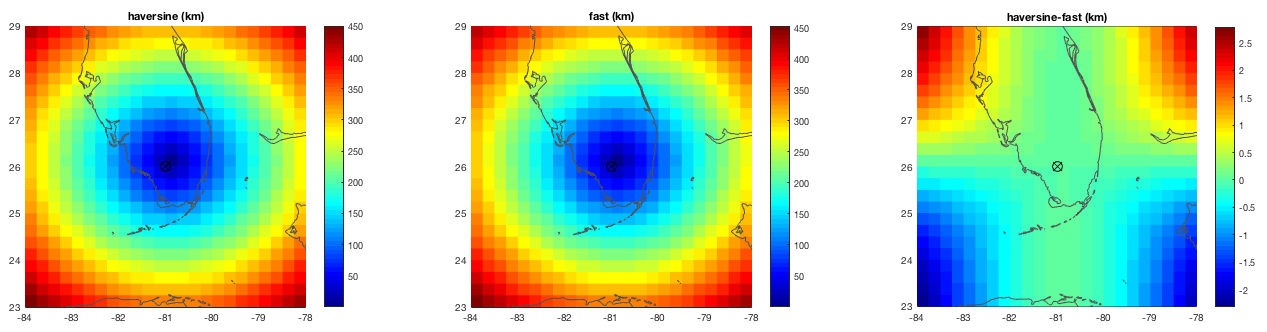


Figure: distance in km between grid points and the reference point (black circle with x), for haversine (left), fast simplified (center) and the difference (right).

Ad b) while isimip calculates the wind field at all centroids for each time step (node) and then stores the maximum wind encountered at each centroid over the lifetime of the cyclone (outer loop over nodes), the default climada routine first figures which time step of the track (node) is closest to a particular centroid and the evaluates the wind field of this node for this centroid (outer loop over centroids). To further save time, climada does ‘fill’ the eye of the cyclone with the maximum wind speed, since due to forward movement, the area within the eye does sooner or later anyway experience the maximum wind speed (of the eyewall). These two approximations do speed up calculations by up to a factor of ten (the larger the factor the more centroids are of concern and the longer the track). For Andrew, for example, the wind field calculation is faster by a factor of 9.5, but this depends on distribution of centroids and number of track nodes (the longer the track and the larger area the centroids are defined for, the larger the factor likely becomes).

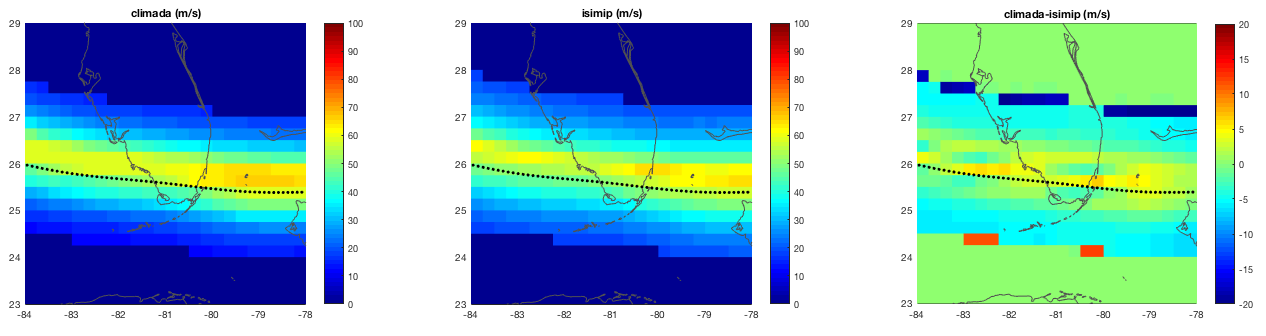


Figure: the climada default wind field (left), the isimip implementation in MATLAB (center) and the difference (right). The large differences at the ‘rim’ are due to different cut-off distances of the wind field calculation and could easily be suppressed. Note that at the core, differences are surprisingly small, given the substantial difference in calculation (inner versus outer loop, see text).

Finally, we compared the isimip implementation in MATLAB with the original python code. Differences are again small, as the following figures illustrate.

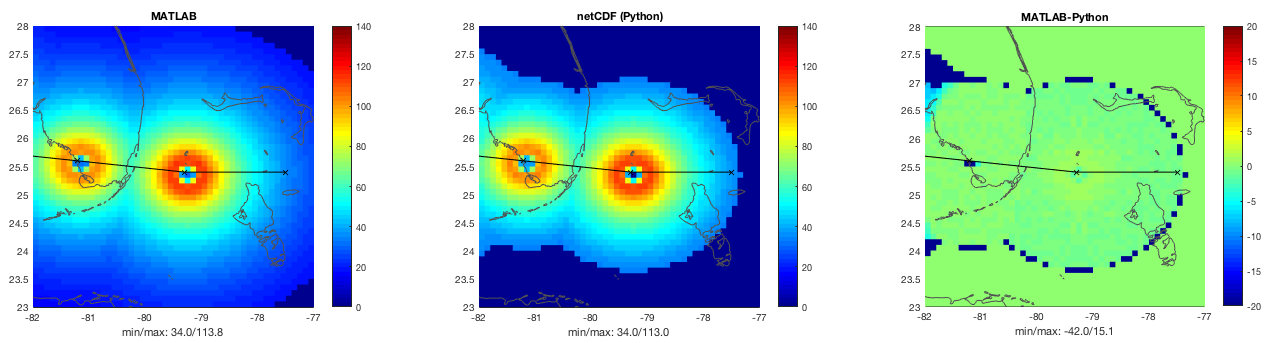


Figure: The symmetric Holland wind field calculation in MATLAB (left), python (center, read from netCDF) and the difference (right), all for just two nodes, 6h apart (Andrew, 1992). Note that MATLAB stores double precision, while the netCDF is uint16, hence the differences.

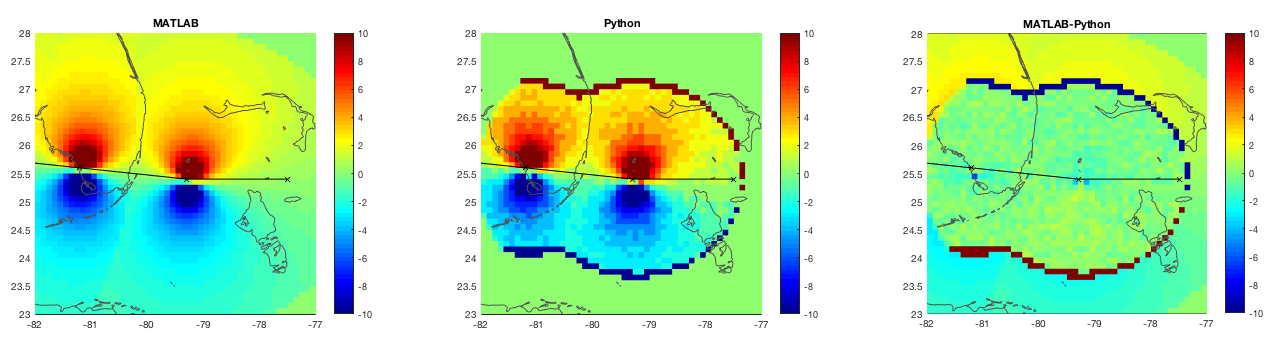


Figure: The translation wind speed (vtrans) alone (Andrew, 1992, just two nodes, 6h apart). MATLAB (left), python (center, read from netCDF, taken as a difference between two fields, hence the ‘rim’) and the difference (right). Note that MATLAB stores double precision, while the netCDF is uint16, hence the ‘clutter’ in the differences.

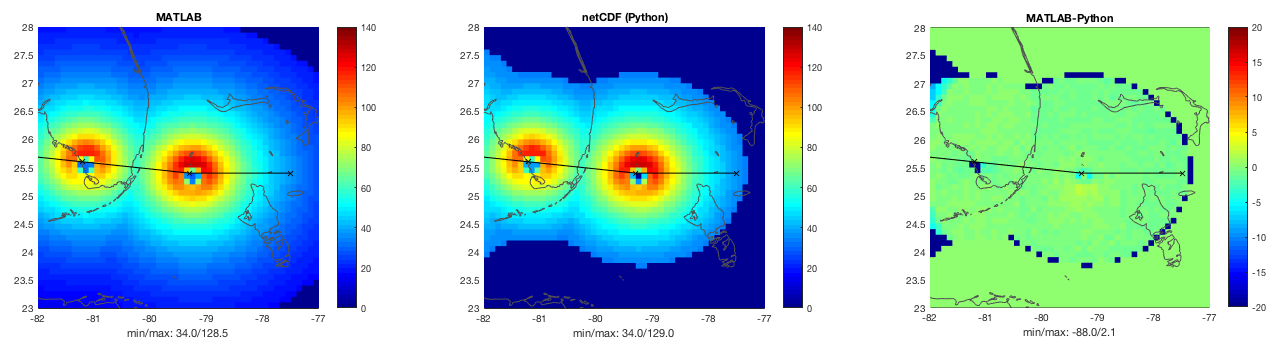


Figure: The full cyclone wind field (Andrew, 1992, just two nodes, 6h apart). MATLAB (left), python (center, read from netCDF) and the difference (right). Note that MATLAB stores double precision, while the netCDF is uint16, hence the ‘clutter’ in the differences and the ‘rim’ again due to the fact that slightly different outer limits of the wind field were applied, as can be seen in the wider (light blue) color of then leftmost pane. The larger differences at the left corners of the rightmost lot are due to the fact the MATLAB implementation did not calculate the next node (to the West).

Next, we will compile the full historic set of tropical cyclone wind fields (so-called historic hazard set) based both on the isimip wind field as well as on the fast default climada approach.

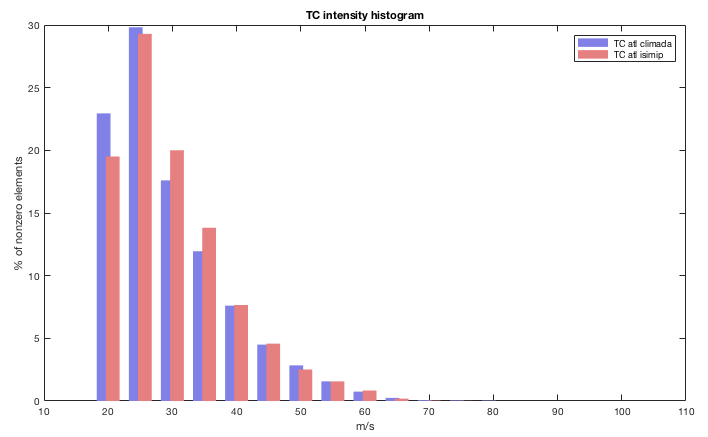


Figure: The histogram of the wind speed values > 20 m/s. climada in blue, isimip in red. The larger number of low wind speed elements for climada is due to the somewhat larger distance to center of the cyclone at a given time step the wind field is calculated for.

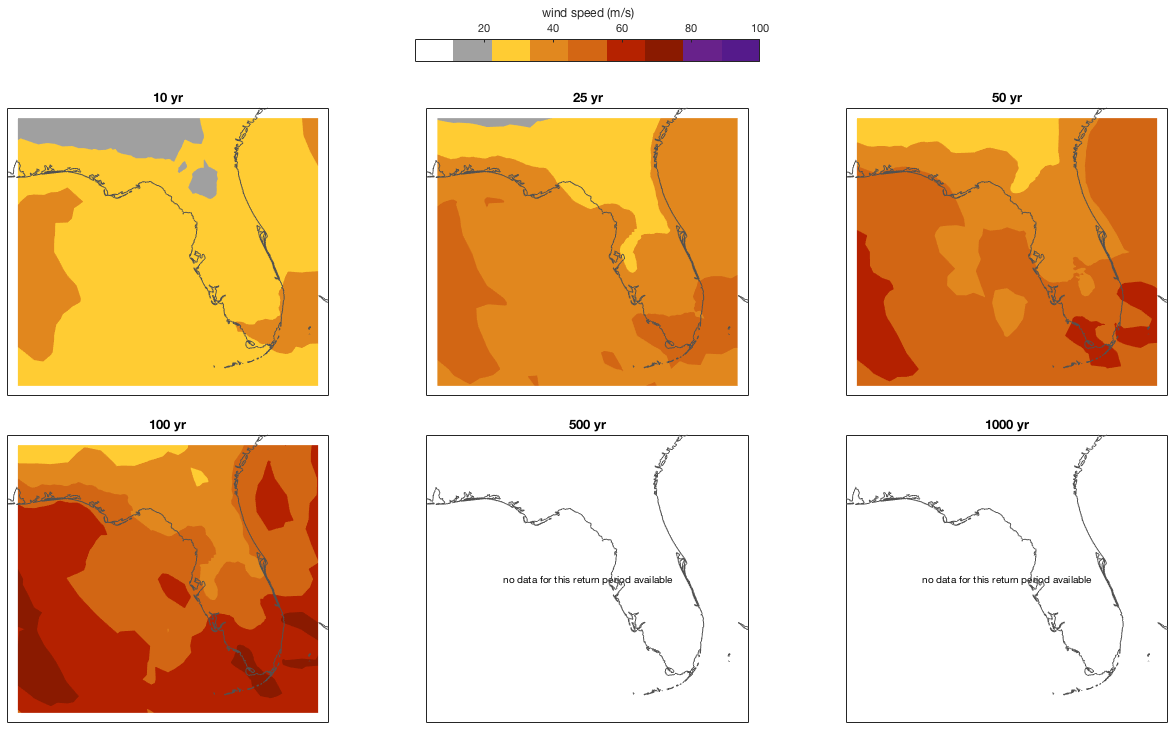


Figure: The wind speed intensity maps for some select return periods based on climada default calculation for all historic tropical cyclones in the North Atlantic. Note that due to the limited time series of historic events, 500 and 1000 years return period maps remain empty.

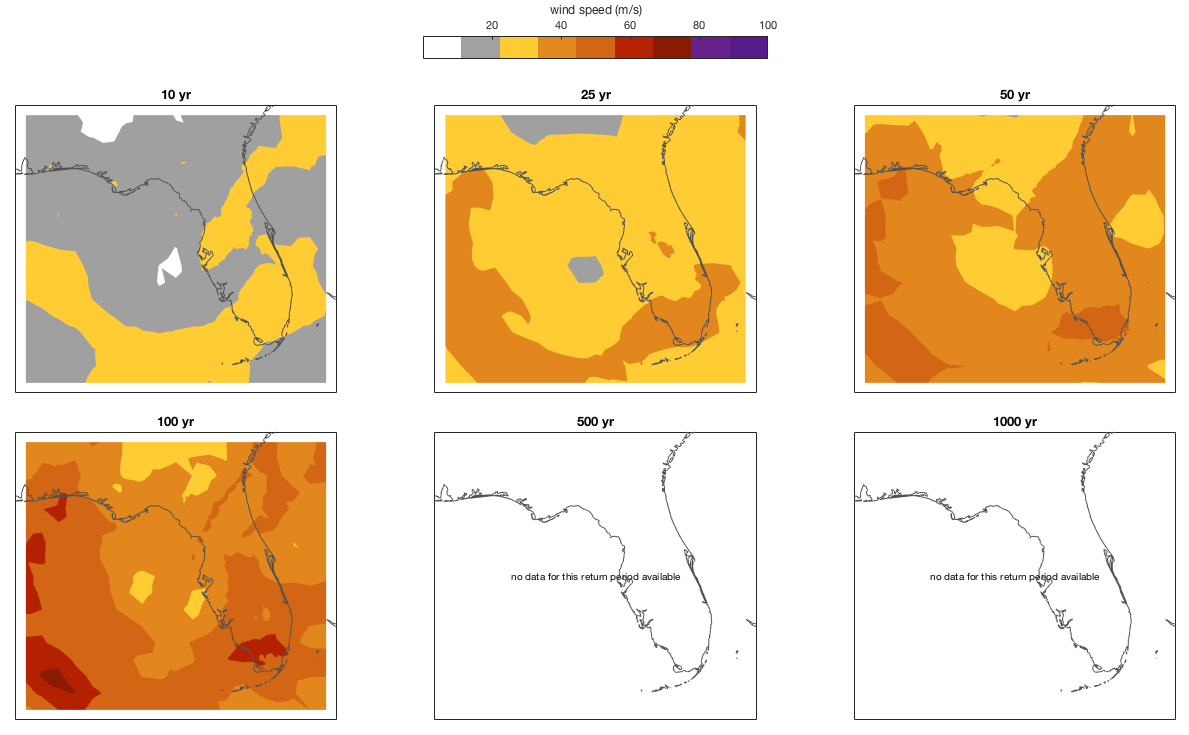


Figure: As above, the wind speed intensity maps for some select return periods, but based on isimip wind field calculation.

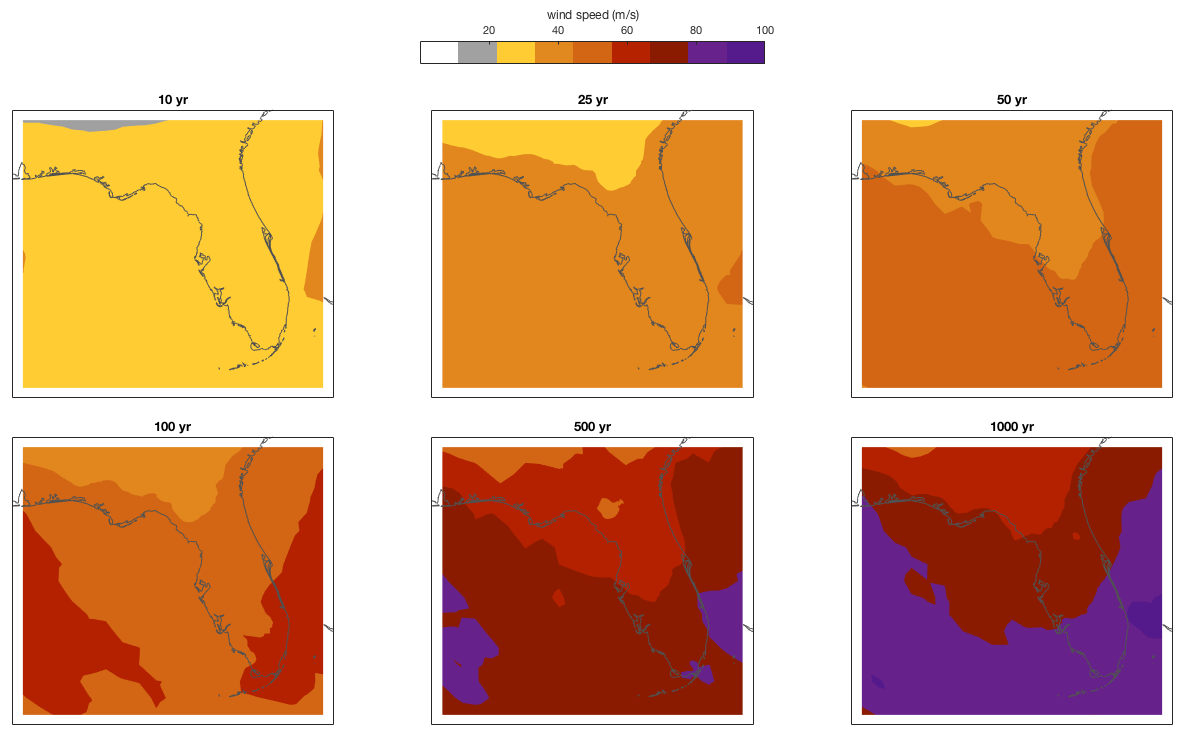


Figure: As above, but for the probabilistic set (14’000 events) for climada standard wind field calculation.

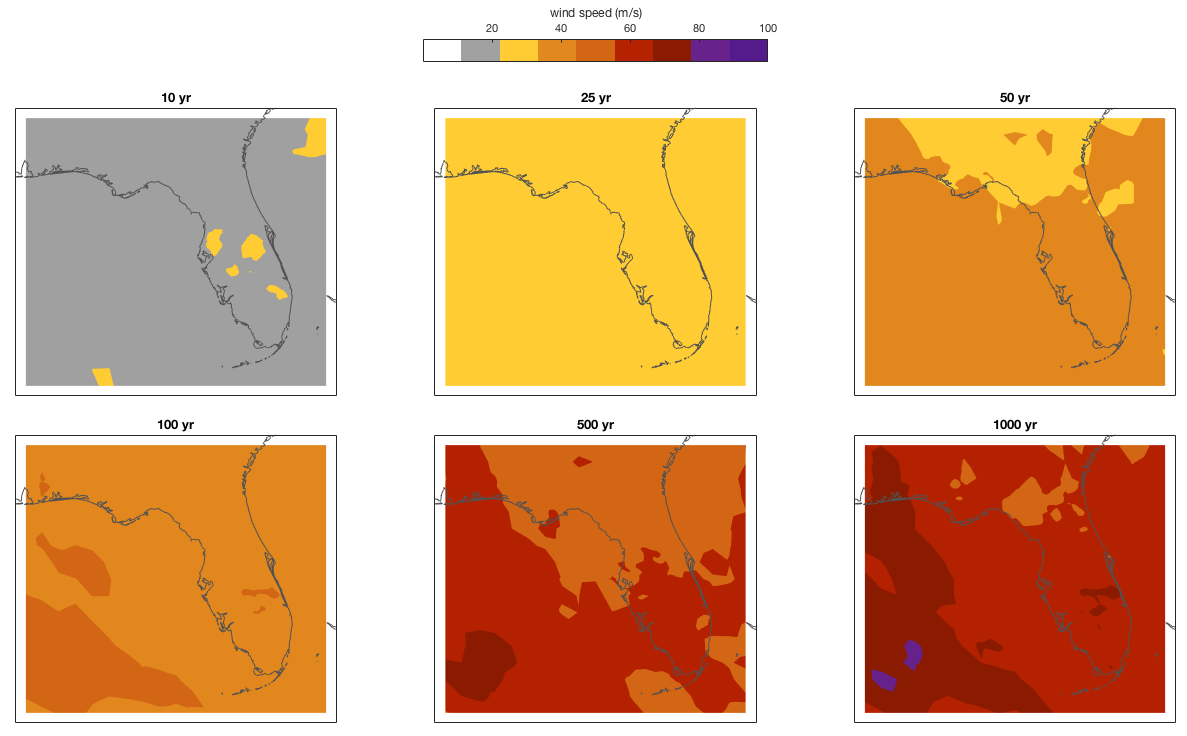


Figure: As above, but for the probabilistic set (14’000 events) for isimip wind field calculation.

And finally the histogram comparison:

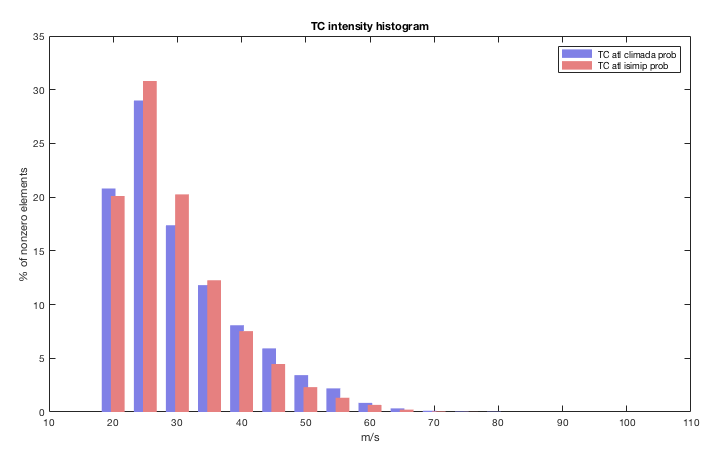


Figure: The histogram of the wind speed values > 20 m/s. climada in blue, isimip in red for the full probabilistic set.

To conclude, both approaches lead to similar results, the climada calculation being just substantially faster – a fact that does matter for users with limited resources, but not for those with access to a cluster.

1. See the isimip module for climada at <https://github.com/davidnbresch/climada_module_isimip> [↑](#footnote-ref-1)
2. For details about climada, see <https://github.com/davidnbresch/climada> and the manual in particular at <https://github.com/davidnbresch/climada/blob/master/docs/climada_manual.pdf> [↑](#footnote-ref-2)
3. For distances on a sphere less than 1000km, the error is mall and can be tolerated for the present need, e.g. about 1 km for a distance of 250km. [↑](#footnote-ref-3)