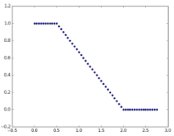
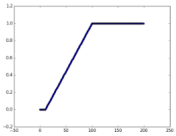
Prototype of non-uniform grid

This is a very short introduction to what I have implemented.

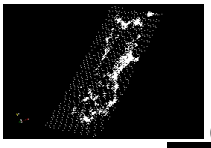
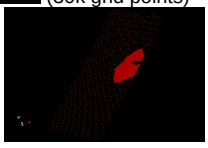
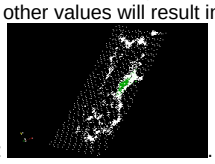
Ideas

- No longer use quad-trees (ie. bisect squares into 4). Instead, we have the following approach:
  - Convert data to grid points
  - Generate the coarsest mesh first (8 Km)
  - Around each point, generate a square domain, half of the distance to the next point
  - Estimate the quality of that square domain. If accepted, add new **6** points. This number is needed to ensure the continuity of the grid as expected
  - Proceed the same way for every point on the coarsest mesh. This will create a new mesh that we will call mesh level 1.
  - Apply the same idea to every point of mesh level 1, resulting into mesh level 2.
  - Stop refining once the distance between points is the smallest acceptable (500 m or other)
  - Convert grid points back to longitude/latitude
- Estimation based around population and vs_500
  - total_score_domain = w_v*score_vs500 + w_p*score_population. w_v + w_p = 1.0
  - Initially w_v=w_p=0.5
  - score_vs500 = 1 if vs_500 < 0.5 Km/s, 0 if vs_500 > 2 Km/s and linearly between 0 and 1 for other values 
  - score_population = 1 if population in domain > k*min_population_acceptable, 0 if population < min_population_acceptable. Linearly valued between 0 and 1 otherwise.  (in the image k=10 just for the example)
  - For vs_500 we use the minimal value of vs_500 found in the tentative domain
  - For population, we use the max value of population found in the tentative domain
  - Arbitrarily: if total_score_domain > 0.5 then refine the region. We can change this value, making it smaller will accept lots more of points
  - ~~Other idea: use mean instead of max for population.~~

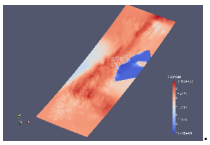
Results

We consider a grid over an 850km x 300km grid around the South Island. If we have a uniform grid. A mesh with 500m spacing for this domain would contain 1701x601 = 1022301 points. In the following we will compare to that quantity.

Make the coefficients w_v and w_p vary and discuss the resulting non-uniform grids.

- w_v = 1.0 gives us a population based grid only:  (30k grid points)
- w_p = 1.0 seems to give a fine grid only based around Canterbury:  (80k grid points)
- Any other values will result in grids in between those extremes. For example, if w_v=w_p=0.5, we get the green points on top of the population grid:  . Here, the green points are the supplementary points added to the pure population grid.

Remarks

- The colored map of the vs_500 values looks like:  .