Loop and Cross for Pulse 73

P. McGill

12Jan21

Find the SES's location from ranges obtained by Kent's new Loop and Cross code on the Wave Glider on 18Jun21 during Pulse 73.

The range of the Rover as detected by the Wave Glider is the straight-line (Euclidean) distance from the Wave Glider's location on the surface to the target's lat, lon, and depth on the sea bottom. Define a function to find the mean squared error between a proposed lat, lon, and depth, and all the ranges and Wave Glider positions given in a list of ranging hits. Locations are given as east, north, up (ENU) units.

```
locDepthErrorENU[hitList_, target_, depth_] :=
 Sum[
   ((EuclideanDistance[ (* find the distance between... *)
        Append[hitList[[k, 1]], 0], (* the Wave Glider position on surface *)
        Append[target, depth] (* and proposed target position and depth *)
       ] (* subtract the measured range and square the result *)
       - hitList[[k, 2]])^2),
   {k, 1, Length[hitList]} (* iterate over the whole list *)
  ] / Length[hitList] (* find the mean *)
```

Import the data.

```
rawdat =
  Import["/Users/mcgill/Documents/My\ Projects/Pelagic-Benthic\ Coupling/Loop\
     and\ Cross/lc-rangeutm-20210618-221923.csv", "CSV"];
```

The column labels are in the first row, and the rest of the data follow.

```
First@rawdat
```

```
{⌗ session[20210618-221923] format: [time (epoch s), context, UTM_easting,
 UTM_northing, UTM_zone_number, UTM_zone_letter, range (m), distance (m)]}
```

```
rawdat[[2 ;; 5]]
```

$$\{\{1.62406\times10^{9}, \text{'DETECT'}, 499\,699., 3.89105\times10^{6}, 10, S, 3584.1, 3549.05\},$$
$$\{1.62406\times10^{9}, \text{'DETECT'}, 499\,856., 3.89078\times10^{6}, 10, S, 3498.4, 3462.49\},$$
$$\{1.62406\times10^{9}, \text{'DETECT'}, 500\,011., 3.89051\times10^{6}, 10, S, 3436.6, 3400.03\},$$
$$\{1.62406\times10^{9}, \text{'DETECT'}, 500\,179., 3.89022\times10^{6}, 10, S, 3404.3, 3367.38\}\}$$

Format the list of ranges and the locations from which they were taken. We want each row to be {{ea-

sting, northing}, range}.

```
hits = Transpose[{rawdat[[All, 3 ;; 4]], rawdat[[All, 7]]}];
```

```
hits[[1 ;; 5]]
```

```
{{{UTM_easting, UTM_northing}, range (m)},
 {{499 699., 3.89105 × 10^6}, 3584.1}, {{499 856., 3.89078 × 10^6}, 3498.4},
 {{500 011., 3.89051 × 10^6}, 3436.6}, {{500 179., 3.89022 × 10^6}, 3404.3}}
```

Drop the labels in the first row.

```
hits = Drop[hits, 1];
```

Find the plot range by finding the range of the hitlist and then adding a margin.

```
eMin = Min[hits[[All, 1, 1]]]
```

```
498 634.
```

```
eMax = Max[hits[[All, 1, 1]]]
```

```
500 861.
```

```
nMin = Min[hits[[All, 1, 2]]]
```

```
3.889 × 10^6
```

```
nMax = Max[hits[[All, 1, 2]]]
```

```
3.89105 × 10^6
```

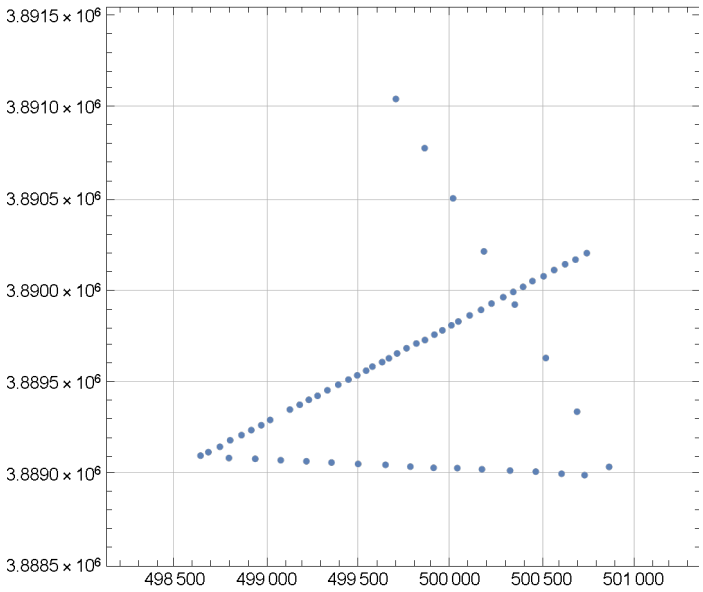
Add a margin to the min and max lats and lons to establish a plot region.

```
margin = 500;
pr = {{eMin, eMax}, {nMin, nMax}} + {{-margin, margin}, {-margin, margin}}
```

```
{{498 134., 501 361.}, {3.8885 × 10^6, 3.89155 × 10^6}}
```

Plot the list of locations from which WG *Tiny* obtained ranges.

```
hitPlot = ListPlot[hits[[All, 1]], PlotRange → pr, GridLines → Automatic,
  AspectRatio → Automatic, Frame → True, ImageSize → Medium]
```



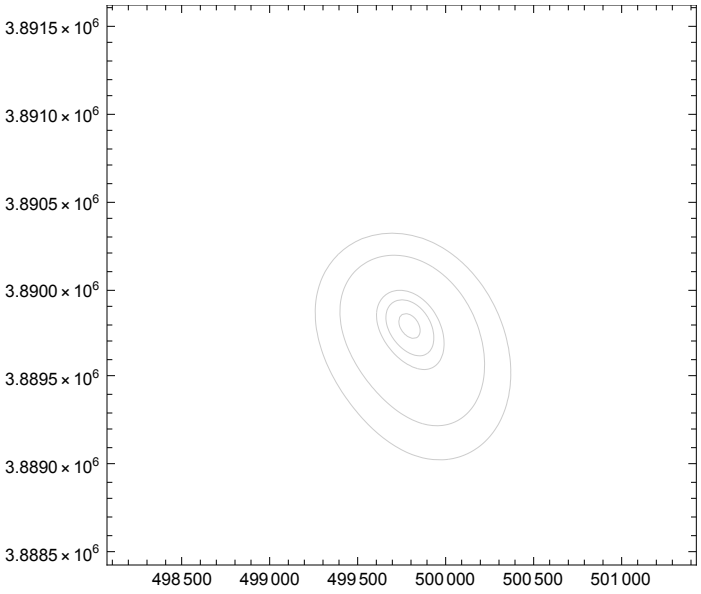
Find the location and depth which minimizes the MSE when compared to the recorded ranges. Note that only the InteriorPoint method can be used with constraints, and I'm setting the constraints to be the edges of the plot. If the best solution is outside the ranges of the plot, this method will fail to find it. Sometimes the MaxIterations will have to be increased for the best solution to be found before the algorithm gives up.

```
sol = FindMinimum[
  {locDepthErrorENU[hits, {east, north}, depth], pr[[1, 1]] < east < pr[[1, 2]],
   pr[[2, 1]] < north < pr[[2, 2]], 3000 < depth < 4500}, {east, north, depth},
  Method → "InteriorPoint", AccuracyGoal → 2, MaxIterations → 1000]
```

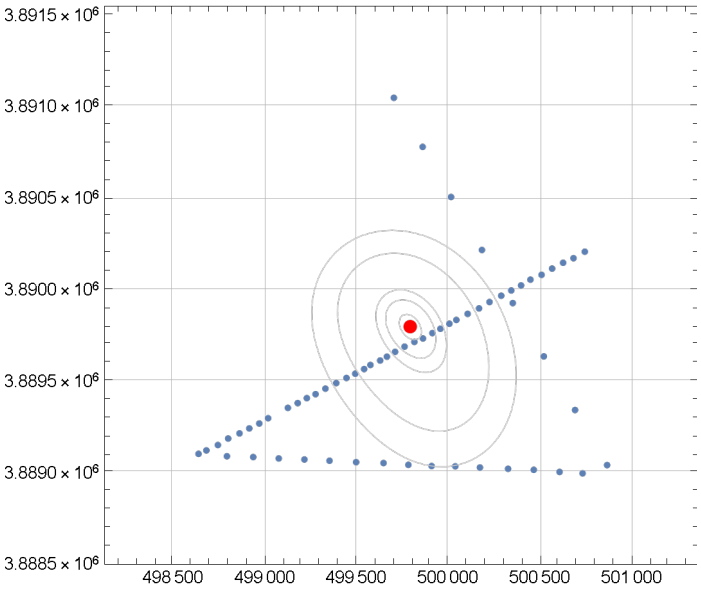
$$\{1.54916, \{\text{east} \to 499\,788., \text{north} \to 3.8898 \times 10^6, \text{depth} \to 3354.69\}\}$$

Plot the target location with a contour plot of the error surface.

```
conPlot = ContourPlot[locDepthErrorENU[hits, {east, north}, sol[[2, 3, 2]]],
   {east, pr[[1, 1]], pr[[1, 2]]}, {north, pr[[2, 1]], pr[[2, 2]]},
   Contours → {100, 500, 1000, 5000, 10 000}, AspectRatio → Automatic,
   ContourStyle → GrayLevel[0.8], ContourShading → None, ImageSize → Medium]
```

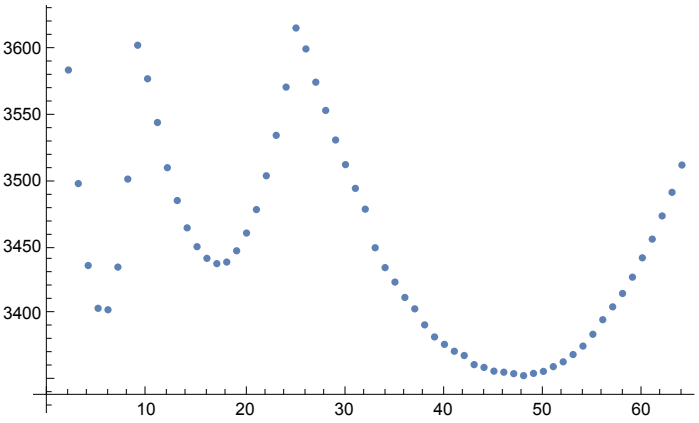



```
Show[hitPlot, conPlot,
 Graphics[{Red, PointSize[Large], Point[{sol[[2, 1, 2]], sol[[2, 2, 2]]}]}]]
```



Plot the ranges as a function of ping number.

```
ListPlot[rawdat[[All, 7]]]
```



Convert the SES drop point from LatLong to UTM and add it to the plot.

```
dropLocSES = GeoGridPosition[
  GeoPosition[{FromDMS[{35, 9.0749}], FromDMS[{-123, -0.2655}]}], "UTMZone10"]
```

GeoGridPosition[{499 597., 3.88982 × 10^6}, UTMZone10]

```
Show[hitPlot, conPlot, Graphics[{Red, PointSize[Large],
    Point[{sol[[2, 1, 2]], sol[[2, 2, 2]]}], Green, Point[dropLocSES[[1]]}]]
```