**LDP Design Work Flow** v.1.0 draft

1. **Identify Area of Interest**

Target the primary population and infrastructure (where will it need to work?)

1. AK road system
2. Public Airports
3. Incorporated city boundaries
4. Borough boundaries
5. Populated places
6. ANCSA Corporations
7. Borough Tax Parcels
8. RS 2477 trails
9. Major Lakes
10. Major Rivers
11. Swamps, wetlands (areas unlikely to be developed)

Where are any adjoining zones, and where is the new area needing to actually work? Load the relevant data into the GIS model and identify the area for the LDP you intend to design.

1. **Model the Ellipsoid Heights**

Create a model of the ellipsoid height = elevation + geoid (.tiff) digital terrain model

1. Crop a large section to cover the area of interest, usually 10-20km larger than expected area
2. Save to work area
3. Spot check model with known points (NGS, PAC/SAC, Survey points, other control, etc.)
4. This is a sub model of the overall model for the state, a local cropping

Will save this and use in next step. Might find you did not get large enough area later on, or it is way larger than desired. But can edit the next product down if needed.

1. **Build Ellipsoid Height grid**

Essentially just making a grid of vector points from the raster model in step 2

Lat, long, height (meters) format has to be: decimal.degrees .csv

1. Spacing is 100m, 150m, 200m etc. have to choose
2. Or can by arc seconds, don’t expect it to make an evenly spaced grid
3. Depending on spacing is if this is pixelated later. We don’t care right now.
4. Save to work area

This is used to compute distortion on a large gridded area, to check the performance of whatever projection design is being looked at.

1. **Create Design Points**

These are the points that a projection will be best fit to. Lat, long, height, pt# as .csv (decimal.degrees, meters)

1. Human decision on what points, and where, to use
2. Spread through area, located where development is expected to occur

(not on mountain tops or bottom of lakes or middle of swamps!)

1. Weight areas by adding more points
2. Might edit the point list after seeing results, by adding or removing
3. Create points by selecting from the ellipsoid model

(can use Global Mapper)

1. **Optimize Projections**

Fit the three projection types, TM, LCC, and OM, to the design points to get a best fit. Compares the variance and range of distortion per design point per projection parameter and finds solution to minimize. Finally minimizes the average toward zero. Uses brute force, except OM routine makes some smart decisions in order to speed it up.

* 1. LDP\_Designer.py (main script, not made yet. Will call other scripts.)
  2. GUI
  3. User selects the design point file made in step 4
  4. Can select to run TM\_Optimizer.py, LCC\_Optimizer.py, or OM\_Optimizer3.py
  5. Each optimizer script computes distortion at points, by calling LambertConformalConic.py, TransverseMercator.py, and ObliqueMercator.py
  6. Should allow some *settings* (use variance, or use range, or both.
  7. Adjust the steps (but suggest defaults)
  8. Save results to prj file, user selects location and name.
  9. We need some naming conventions, and folders named etc. so we do it the same way

1. **Evaluate Projections**

Check the results of step 5 by making a simple distortion map.

* 1. Distortion\_Writer.py (LambertConformalConic.py, TransverseMercator.py, and ObliqueMercator.py)
  2. GUI
  3. User selects the gridded vector values from step 3
  4. Computes the distortion as a ppm then write the point to a new fil

(lat, long, height, distortion(ppm)

* 1. Save answer as asci file. Probably .csv use place to save and name convention
  2. View results in a GIS (color coding the distortion attribute by value (20ppm = good)
  3. View design points, they might contribute to a poor answer and need edited

If so, edit design points and re-run step 5 and 6

* 1. Suggest which projection is best for area (human decision)

1. **Define Zone Extents**

While inspecting results in step 6, draw polyline around area where it works according to the design constraints.

* 1. Save results, probably name the shapefile by
  2. User selects the gridded vector values from step 3
  3. Computes the distortion as a ppm then write the point to a new fil

(lat, long, height, distortion(ppm)

* 1. Save answer as asci file. Probably .csv use place to save and name convention
  2. View results in a GIS (color coding the distortion attribute by value (20ppm = good)
  3. View design points, they might contribute to a poor answer and need edited

If so, edit design points and re-run step 5 and 6 (human decision)

* 1. Eventually overlapping areas will be trimmed to a final zone location.
  2. Also areas where two zones barely did not meet, will be extended and assume some distortion.
  3. Goal is always largest zone possible that is also in harmony with overall zones.
  4. Considering the population and infrastructure in step 1
  5. Cannot always fit to all of it, so pick best of both worlds (good LDP and good zone)

**Known stuff to work on:**

LDP\_Designer.py needs to be written, made GUI. Add some helpful text.

LambertConformalConic.py needs edited, not correct values on double parallels

TransverseMercator.py, and ObliqueMercator.py need written. These are pulling the computational power out of the current script “akdot\_nr\_ldp\_v1.py” which is being phased out.

Distortion\_Writer.py needs edited to use generic projection values, and call LambertConformalConic.py, TransverseMercator.py, and ObliqueMercator.py. Also make it GUI.

TM\_Optimizer.py, LCC\_Optimizer.py, or OM\_Optimizer3.py all need edited to use the LambertConformalConic.py, TransverseMercator.py, and ObliqueMercator.py. Also need to let user select the ‘steps’ to control how thorough the brute force works. Have defaults, and suggest setting. Also Optimizers need to allow to use the RANGE instead of the VARIANCE. Or run both. Michael Dennis is suggesting Range is what works for him. It is not clear which is better, and probably totally based on what the human did in step 4. We might as well code in this as an option. I know what to do. And the results in step 6 are what drives the final design used.

As a final note to myself, need to have optimizer automatically clip the results to nearest 10 seconds, or nearest minute, or so. Similar for the scale factor, clip results to appropriate decimal place.