TERRA – Setting Up Flights

Included in git repo

# Creating the Box

1. Load data from one flight.
2. Check flight track by displaying latitude vs longitude.
3. Run TERRAInstall.cmd. This should add the appropriate files to your user procedures directory.
4. Load FlightFitting.ipf in your Igor experiment (#include “FlightFitting”).
5. Load flight path csv file (this can be generated by GIS or other means outside of Igor) and plot it with the flight track to ensure everything looks correct.
6. Create an excel file with the start and end times of your boxes matching Example\_BoxDetails\_2018.xlsx. Load this file into your Igor experiment.
7. Run getIndex (BorS, Name, Time) where BorS = “box”, Name = the name of the box (in BoxName column – Example\_BoxDetails\_2018.xlsx) and Time is the date/time wave of the data to get the start and end index of the box (stIndex, endIndex).
8. If there are parts of the flight that occur between the start and end time of the box but should not be included in the kriged screen (e.g. spirals) create exc\_st and exc\_end containing the start and end time for each section that should NOT be included (refer to Example\_AIRCRAFT\_Box\_Screen\_Spiral\_Times.xlsx as an example).
9. Determine whether flight path goes clockwise or counterclockwise.
10. If it goes clockwise, run switchDirection to make the path counterclockwise.
11. Run updateBoxPath(2000, 500, Lat, Lon, Time) where 2000 and 500 are the size of the ellipse in which the points will be averaged to create a path and Lat, Lon and Time are the corresponding waves containing the latitude, longitude and datetime of the data.
12. Zoom in closely on the corners of the box and look for any areas where the track (fitpath\_1m) overlaps itself. Delete points from fitpath (NOTE different wave name) that are causing overlap and run upSamplePath (Lat, Lon) – For example image see file: Example\_path\_overlap.jpg
13. Check that the path is a good fit of the flight track – you can test different values in the updateBoxPath function in place of 2000 and 500 and see if this improves the fit. Once you have selected values to move forward with enter the first value into the Example\_BoxDetails\_2018.xlsx or equivalent spreadsheet under “Ellipse perp” and the second value under “Ellipse parallel”.
14. Get the index of the southeast corner (the wave name is fitpath\_1m) of the box and run Shift(offset, Lat, Lon) with that value to place the start of the box in the southeast corner.
15. Get the indices of the 4 corners of the box and replace the values in MakeFrame to reflect these multiplied by 2 and run MakeFrame(SE corner, NE corner, NW corner, SW corner, SE corner end) where SE corner = 0, NE corner = index of NE corner, NW corner = index of NW corner, SW corner = index of SW corner, SE corner end = index of last point on the path (*for a 5 sided box use MakeFrame5(SE, NE, NW, W, SW, SE corner end) where W is the corner on the west side of the box*).
16. Obtain a digital elevation model for the study area.
17. Adjust load2() SetScale commands to fit with the digital elevation model being used.
18. Run loadElevations() to load in the digital elevation model and scale it, and to get the ground elevation below the flight track and below the best fit flight box/screen – this will take a few minutes.

# Creating Fit of Wind Below Flight Track

1. Open WindProfiles\_2018.pxp and load flight start and end times into the experiment.
2. Run MakeProfiles to compile wind data points below the flight track to the ground for each box/screen.
3. Run FitProfiles to fit wind points to a log function.
4. Save fita, fitd, fitb, fitrms (with flight start and end times) to a file.
5. Optional - View profile for a flight by running displayFit(fNum) in WindProfiles\_2018.pxp.

# Create the Wind Screens

1. Run MapPosition2Screens(flightNum, Lat, Lon, Alt, Time) to map the flight points to the best fit flight box– this will take a few minutes.
2. Run Variables(flightNum).
3. Run Flag(fltStr) where fltStr is a text flag to be used to denote which points are part of the box (e.g. “F19\_Syncrude”).
4. Run Wind (fltStr, WindSpeed, WindDirection).
5. Run runAll() and select PositionSZWE from the pull down list to krig the east wind screen – this can take several hours to run.
6. When kriging is complete examine the plot that is generated to see that it makes sense and it looks like kriging has properly interpolated the data values shown as points on the plot.
7. Run runInterp() and select PositionSZWN from the pull down list to krig the north wind screen – this could take a few minutes to run.
8. As before, when kriging is complete examine the plot that is generated and check that it makes sense.
9. Run FillWind(fNum) to use the wind profile data to fill the area below the flight path to the ground.

# Create the Air Flux Screen

1. Run AirDensity (fltStr, Pressure, Temperature, DewPointTemperature).
2. Run runInterp() and select PositionSZA from the pull down list to krig the air density – this could take a few minutes to run.
3. Run FillAir() and check the plot.
4. Run FluxCalcSetup() to get the air flux through the screen.
5. Create folder for the flight using the name = fltStr.
6. Run exportTERRAWaves(fltStr) to export the files as Igor binary and choose the created folder.
7. Save experiment.
8. Open a new experiment file and use “Browse expt...” on the Data Browser to link to the previous file.
9. Copy over the weights and weightsLoc waves to the new experiment and save the experiment as weights\_fltStr (for example weights\_F19) to match the other files.

# Calculate the Box Area and Mass Emission

1. Open MassEmissions\_2018.pxp and load box start and end times.
2. Run getTempPress(boxName) to get starting and ending temperature and pressure values.
3. User will be prompted for the location of the files they exported in step 37 (choose the folder one level above the specific flight folder).
4. Run calcEairm().
5. Run exportMassEmisWvs(fltStr).