1 Python Code

To use the functionality of this code the scipy stack is required. To enable writing of LaTeX files MikTex needs to be installed. If further Latin hypercubes need to be calculated MATLAB needs to run on the system. Calls to MikTex and MATLAB are wrtten on a windows computere so the cammand line calls are in windows syntax.

The python code included within the CD has the following modules:

Shared is a module that contains all the logic available that is required by all modules of the project and contains generic functions that could be used in all locations.

Plane energy is a module that is used to model the energy consumption of a plane. This module contains a number of global variables that define a default plane. A single class within this module is constructed using plane variables, this class is capable of computing the energy required for a number of flight manoeuvres and returning the energy coefficient and energy factor.

Airfoil is a module that is used to obtain data on a number of airfoils using *airfoiltools.com*. The class contained within this module is constructed with a foil name and Reynolds number and can return the variation of the lift and drag coefficients with changing angle of attack.

Latin hypercube is a module that calls MATLAB to connect with the code produced by Forrester et al., 2008 and return Latin hypercube sampling plans of any given number of nodes and number of dimensions. The results to these MATLAB calls are cached to reduce the time of subsequent calls.

Travelling plane is a module that is used to calculate the least cost route through given nodes. The module contains both the exact all routes approach and the progressive travelling plane approach.

Sample model is a module that is used to compute models of the energy cost of a routes given different scenarios. This module enables the route planning for atmospheric data collection to be done from the requirement of total energy consumed.

Dubins path is a module used to compute the shortest distance Dubins paths either between two points with start and end directions defined or using the Dubins Path class the total path through a number of nodes. This module is not fully commented as the geometric logic is from Giese, 2012 and there is a lot of lines of code to explain.

2 References

[Forrester et al., 2008] Forrester, A., Sóbester, A., and Keane, A. (2008). *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons.

[Giese, 2012] Giese, A. (2012). A comprehensive, step-by-step tutorial on computing dubins curves.