WingLoft - A Simple Wing Lofting Program

John Nappi – Introduction to C++

Due Date: April 1, 2019

Brief

WingLoft is a simple but powerful wing lofting program with outputs for 3D plotting in Matlab or a VTK Viewer (e.g. Paraview).

Build Instructions

The solution can be built in VisualStudio in two different ways:

For Running Unit Tests:

1. Right click WingLoft project, click Properties, and set Configuration Type to "Static library (.lib)".
2. Right click WingLoftTester project and "Set as StartUp Project"
3. Build and run the unit tests.

For Running WingLoft as an Application:

1. Right click WingLoft project, click Properties, and set Configuration Type to "Application (.exe)".
2. Right click WingLoftproject and "Set as StartUp Project"
3. Build.
4. Application can be run from debugger (arguments are set to read the CRM Wing and output both .vtk and .m file), or run from the Window's command prompt

How to Run

Running WingLoft.exe with no arguments provides instructions for program usage:

```
Illegal number of arguments. Usage:

WINGLOFT inputfile [-v] [-m]

  inputfile  Specifies the wing planform file.  If full path not
             specified, file is assumed to be in working directory.
  -v         Outputs wing loft in a .vtk file with same basename as
             inputfile.
  -m         Outputs wing loft in a .m file with same basename as
             inputfile.

At least one of -v or -m must be present, but both can be specified.
```

Example

The following reads the CRM Wing planform and outputs both a matlab file and vtk file for plotting:

```
WingLoft ".\input\CRM Wing\CRM_Planform.dat" -m -v
```

Wing Planform File

The wing planform file is a text file describing the wing planform. With the exception of twist, the file is units agnostic – the only requirement is that the values are consistent. The structure of the file is as follows:

Line1 – Title string

- The title is not used by the program, it is solely for the benefit of the user.

Line 2– Number of chord stations

- This parameter affects the chordwise discretization of the wing. Both upper and lower airfoil points are generated at each chord station (except for the leading edge). The chord stations are spaced using the Cosine Rule.

Lines 3 - End – Wing station data organized in columns:

- **Column 1** – X location of the leading edge in global space
- **Column 2** – Y location of the leading edge in global space
- **Column 3** – Z location of the leading edge in global space
- **Column 4** – Chord length measured in the projection of the planform onto the X-Y plane
- **Column 5** – Twist of the wing about the quarter chord in degrees. A positive number results in a nose-up twist.
- **Column 6** – Name of an airfoil file to read, or a NACA 4-digit airfoil to generate internally
    - Airfoil files to read are assumed to be in the same directory as the wing planform file.
    - Airfoil files to read should end in ".dat".
    - To generate a NACA 4 digit airfoil, column 6 should read "NACAxxxx" where XXXX is the 4 digit number (e.g. "NACA4412"). If column 6 ends in ".dat", it is assumed to be a file to be read, and NOT an internally generated airfoil (e.g. "NACA4412.dat" is assumed to be a file)
- **Column 7** – Number of spanwise intervals to create between the current station and the next station.
    - For example, the value of 3 in the first station pictured below generates four airfoils at Y = 0.000, 38.558, 77.117, 115.675 for a total of 3 intervals
    - For the last wing station, this value is ignored but is required to be present and can be any value. The picture below shows the value set to "-1" since it visually indicates it's not real data

```
NASA Common Research Wing
25
 904.294      0.000    174.126    536.181    6.7166     CRMFoil.dat    3
 989.505    115.675    175.722    468.511    4.4402     CRMFoil.dat    2
1032.133    173.513    176.834    434.674    3.6063     CRMFoil.dat    2
1076.030    231.351    177.361    400.835    3.0131     CRMFoil.dat    2
1120.128    289.188    177.912    366.996    2.2419     CRMFoil.dat    2
1164.153    347.026    178.886    333.157    1.5252     CRMFoil.dat    2
1208.203    404.864    180.359    299.317    0.9379     CRMFoil.dat    1
1225.820    427.999    181.071    285.782    0.7635     CRMFoil.dat    1
1252.246    462.701    182.289    277.288    0.4285     CRMFoil.dat    2
1296.289    520.539    184.904    263.130   -0.2621     CRMFoil.dat    2
1340.329    578.377    188.389    248.973   -0.6782     CRMFoil.dat    2
1384.375    636.214    192.736    234.816   -0.9436     CRMFoil.dat    2
1428.416    694.052    197.689    220.658   -1.2067     CRMFoil.dat    2
1472.458    751.890    203.294    206.501   -1.4526     CRMFoil.dat    2
1516.504    809.727    209.794    192.344   -1.6350     CRMFoil.dat    2
1560.544    867.565    217.084    178.186   -1.8158     CRMFoil.dat    2
1604.576    925.402    225.188    164.029   -2.0301     CRMFoil.dat    2
1648.616    983.240    234.082    149.872   -2.2772     CRMFoil.dat    2
1692.659   1041.078    243.635    135.714   -2.5773     CRMFoil.dat    2
1736.701   1098.915    253.591    121.557   -3.1248     CRMFoil.dat    2
1780.737   1156.753    263.827    107.400   -3.7500     CRMFoil.dat   -1
```

Airfoil File

The airfoil file is expected to be in the "Selig Format" which is as follows:

Line1 – Title string

- The title is not used by the program, it is solely for the benefit of the user.

Line2 – Number of points on upper curve (N_{upper}), number of points on lower curve (N_{lower})

Line3 – Blank line

Line4 - (N_{upper} + 3) – Points along the upper cuver

- The points are ordered from leading edge to trailing edge in x/c, y/c pairs.

Line (N_{upper} + 4) – Blank line

Line (N_{upper} + 5) – (N_{upper} + N_{lower} + 4)

- The points are ordered from leading edge to trailing edge in x/c, y/c pairs.

A couple notes about the airfoil are worth pointing out:

- The airfoil should be non-dimensionalized by the chord so that the leading edge is at 0.0, and the trailing edge is at 1.0. Likewise the y coordinates should be non-dimensionalized by the chord. The program <u>does not check for this</u>.

- Internally the airfoil is read in and then reordered to go from the trailing edge to the leading edge along the lower curve, and then back to the trailing edge on the upper curve. The leading edge point is not duplicated.

```
CRM Airfoil Section
       129.        129.

   0.00000    0.00000
   0.00014    0.00182
   0.00054    0.00366
   0.00122    0.00546
   0.00216    0.00724
   0.00338    0.00902
   0.00487    0.01077
   0.00662    0.01248
   0.00864    0.01417
   0.01092    0.01584
```

...

```
   0.98977    0.00469
   0.99272    0.00350
   0.99541    0.00232
   0.99784    0.00115
   1.00000    0.00000

   0.00000    0.00000
   0.00014   -0.00155
   0.00054   -0.00306
   0.00122   -0.00457
   0.00216   -0.00605
```

...

```
   0.98977    0.00334
   0.99272    0.00254
   0.99541    0.00171
   0.99784    0.00088
   1.00000    0.00000
```

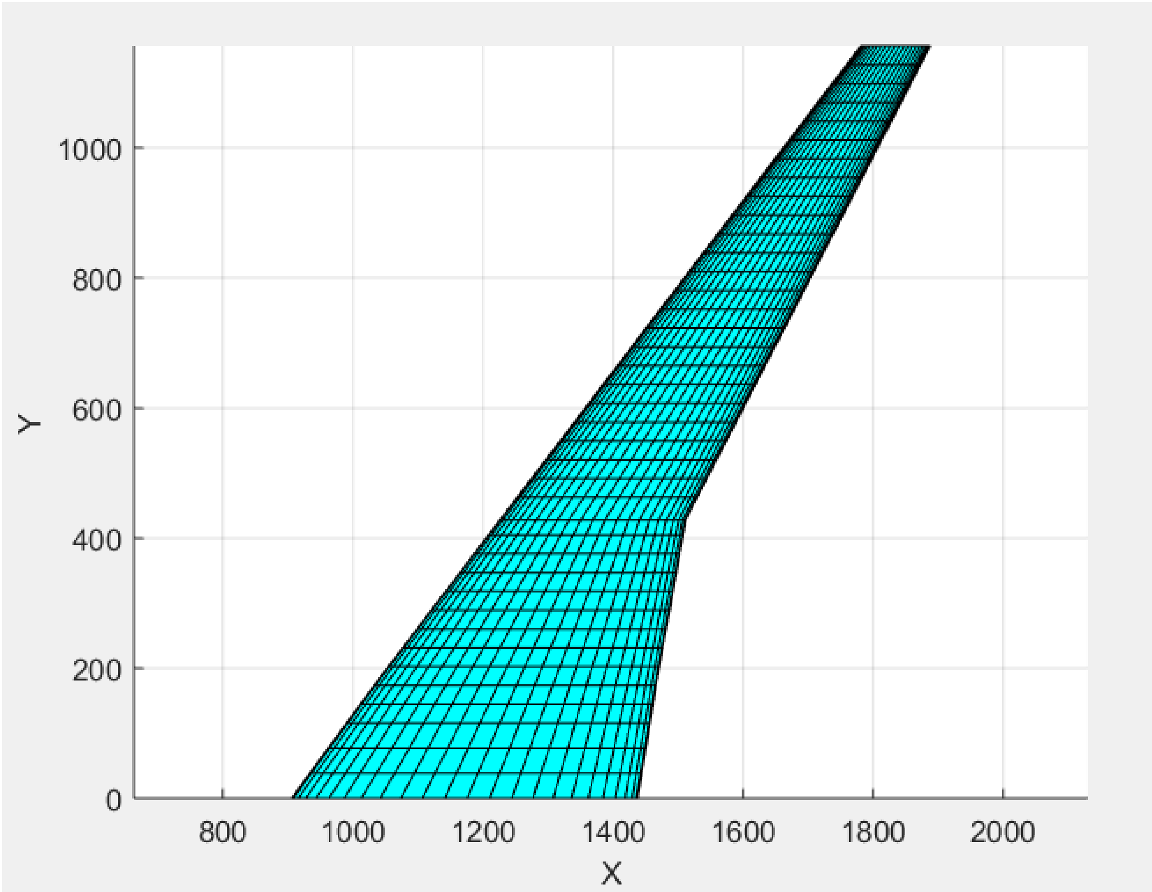
Output

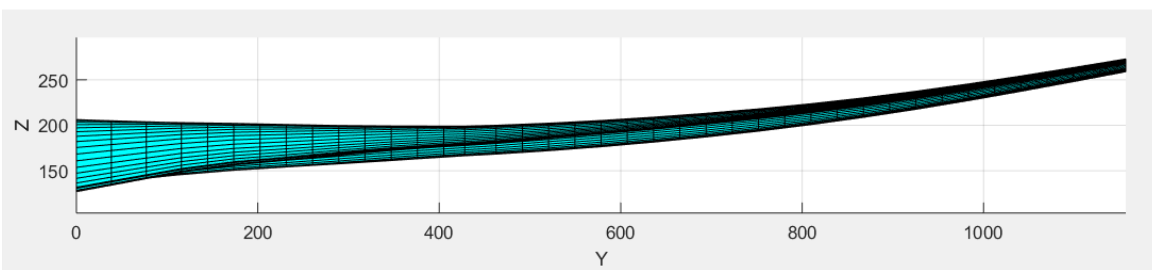
The first available output is a Matlab .m script which, when executed, creates a patch plot. An example of the output file for the NASA CRM Wing, and the generated interactive plot is shown below.

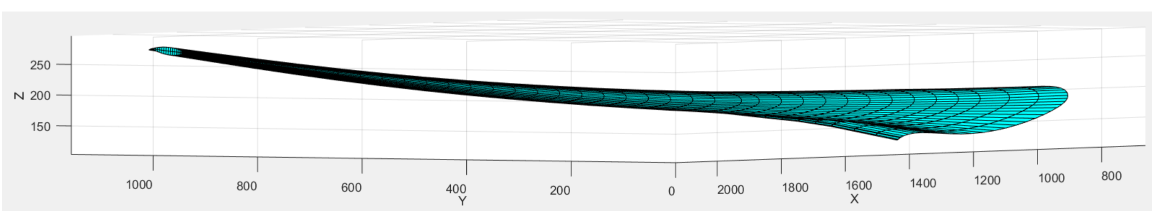
Example of .m file:

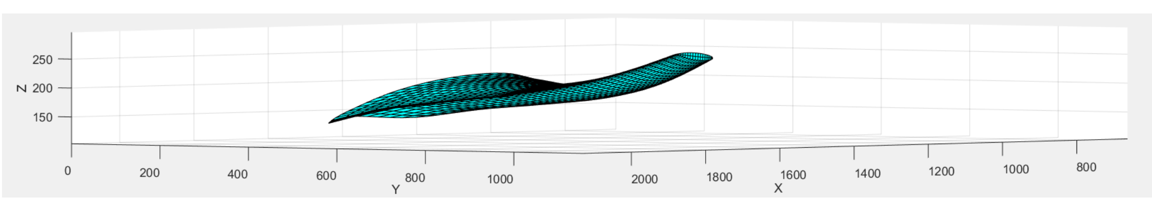
```matlab
% WingLoft Patch Plot
clear; close all; clc;

%% Vertices
V = [
1437.72 0 127.093
1435.54 0 128.215
1428.95 0 130.777
1417.95 0 133.721
...                                   ...
%% Faces
F = [
1 2 51 50
2 3 52 51
3 4 53 52
4 5 54 53
5 6 55 54
...                                   ...
%% Patch Plot
p = patch('Vertices', V, 'Faces', F, 'FaceColor', [0,1,1], 'FaceLighting', 'Gouraud');
axis equal;
grid on;
xlabel('X');
ylabel('Y');
zlabel('Z');
```

Example of Matlab Plot:









Example of .vtk file:

```
# vtk DataFile Version 2.0
FieldData
ASCII
DATASET UNSTRUCTURED_GRID
POINTS 2009 double
1437.72 0 127.093
1435.54 0 128.215
1428.95 0 130.777
1417.95 0 133.721
1402.62 0 136.215
```

...

```
CELLS 1920 9600
4 0 1 50 49
4 1 2 51 50
4 2 3 52 51
4 3 4 53 52
4 4 5 54 53
```

...

```
CELL_TYPES 1920
9
9
9
9
9
```

Example of VTK Plot: