xfoil_polar.m user guide

Lorenzo Frascino - Palma Caputo

Contents

1 Introduction

The function *xfoil_polar.m* allows to evaluate the $C_d - C_l$ polar of an airfoil by using the software *xfoil* [1]. The lift coefficient C_l is obtained by direct surface pressure integration

$$C_l = \int C_p d\bar{x} \tag{1}$$

where the integral is performed in the clockwise direction around the airfoil contour. The pressure coefficient C_p is calculated using the Karman-Tsien compressibility correction.
The drag coefficient C_d is obtained by applying the Squire-Young formula at the last point in the wake. For further information, please refer to [1].

2 I/O

Inputs and outputs are listed in the 2 following subsections.

2.1 Inputs

- **NACA**: number of the NACA airfoil, if the airfoil of interest is NACA. EXAMPLE: having in input '0012' means that the airfoil is NACA 0012;
- **ext_AF**: can be 1 (airfoil geometry is imported from external file) or 0 (airfoil of interest is NACA);
- **numPanel**: number of panels in which the airfoil is divided;
- **Re_number**: Reynolds number;
- **iter**: number of iterations for the viscous calculus;
- **FirstAlfa**: first value of the angle of attack;
- **LastAlfa**: last value of the angle of attack;
- **DeltaAlfa**: pace between one angle of attack and the following one.

2.2 Outputs

- C_l: lift coefficient;
- C_d: drag coefficient.

3 Code description

After having inserted inputs, the code creates an array of angle of attack ALFA_VEC that will then be given to xfoil.
In order not to having overwriting with existing files, it is then checked the possibility and eventually existing files are deleted.

```
Alfa_vec    = ...
    str2num(FirstAlfa):str2num(DeltaAlfa):str2num(LastAlfa);
saveGeometry = 'Airfoil_geometry.txt';  % Create .txt file to ...
    save airfoil coordinates
savePolar  = 'Polar.txt';               % Create .txt file to ...
    save the polar


% Delete files if they exist
if (exist(saveGeometry,'file'))
    delete(saveGeometry);
end

if (exist(savePolar,'file'))
    delete(savePolar);
end
```

The following step is the creation of a .TXT file with commands which have to be passed to xfoil.

```
%% WRITING XFOIL COMMANDS
% Create the airfoil
f_input = fopen('xfoil_input.txt','w');            % Create ...
    input file for xfoil

if ext_AF == 0
        fprintf(f_input,'y\n');
        fprintf(f_input,['naca ' NACA '\n']);
end
if ext_AF == 1
    fprintf(f_input,'y\n');
    fprintf(f_input,'load\n');
    fprintf(f_input, 'my_airfoil.txt\n');          % .txt file ...
        with airfoil coordinates
end

fprintf(f_input,'PPAR\n');
fprintf(f_input,['N ' numPanel '\n']);
fprintf(f_input,'\n\n');

% Data for the polar
fprintf(f_input,'OPER\n');
fprintf(f_input,'visc\n');
fprintf(f_input,[Re_number '\n']);
fprintf(f_input,['iter' iter '\n']);
fprintf(f_input,'pacc\n');
fprintf(f_input,[savePolar '\n\n']);

for i = 1:length(Alfa_vec)

        fprintf(f_input,'a %2.4f\n', Alfa_vec(i));
end
```

```
31
32  fprintf(f_input,'pacc\n\n');
33
34  % Save the airfoil data points
35  fprintf(f_input,['PSAV ' saveGeometry '\n']);
36
37  % Close file
38  fclose(f_input);
```

After having run the software, data file is read and lift and drag coefficient imported from the file. In addition, it is possible to plot the polar in the main with the new-obtained values of C_l and C_d.

```
1  %% RUNNING XFOIL
2  cmd = 'xfoil.exe < xfoil_input.txt';
3  [status,result] = system(cmd);
4
5
6  %% READ DATA FILE
7  filePol = fopen(savePolar);
8  A = textscan(filePol,'%f %f %f %f %f %f %f', 'Headerlines',12);
9  fclose(filePol);
10 alfa = A{1}(:,1);
11 Cl   = A{2}(:,1);
12 Cd   = A{3}(:,1);
13
14 x = find(ismember(Alfa_vec,alfa)==0);
15 for l = 1:length(x)
16     err = sprintf('CONVERGENCE FAILED FOR ANGLE OF ATTACK: ...
           %s\n', num2str(Alfa_vec(x(l))));
17     disp(err);
18 end
```

In order to work, the main and function must be in the same directory as the software xfoil!

4 Test case

The inputs that had been chosen are the following:

```
1  %% INPUT DATA
2  NACA       = '0012';
3  ext_AF     = 0;
4
5  numPanel   = '160';
6  Re_number  = '1e6';
7  iter       = '300';
8  FirstAlfa  = '-15';
9  LastAlfa   = '15';
10 DeltaAlfa  = '0.5';
```

The code provides the $C_d - C_l$ polar in figure 1.

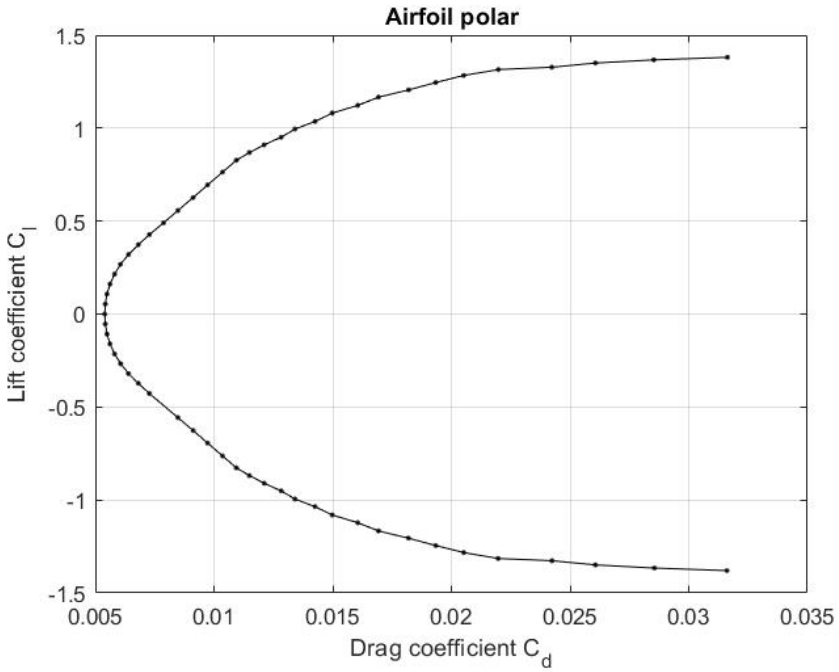

Figure 1: $C_d - C_l$ polar for the test case.

References

[1] *xfoil User's Guide*. MIT. URL: `https://web.mit.edu/drela/Public/web/xfoil/xfoil_doc.txt`.