Project No.2

Optimisation of an airfoil

1. Introduction

The main objective of this project is to optimise the geometry of an airfoil using the gradient method. The results of performed calculations will show the differences in the solutions (in the geometry of the airfoil) resulting from the defined objective function. The exercises are comprised of simple and complex optimisation tasks. In addition, the so-called penalty function (the way of defining it and the effect it causes) will be also analysed.

In each exercise students need to define the objective function and modify the source code accordingly. All exercises are based on the same source code provided by the lecturer. Once the source code is modified it should be compiled and run. Then, the obtained results should be analysed.

2. OptoFoil

OptoFoil software runs the whole process of optimisation and prepares files with results and Xfoil software is used to find the aerodynamic coefficients of the optimised airfoil. A special script is used to connect these two. The scrip is fully prepared by OptoFoil as presented in the following code excerpt:

```
double Xfoil(double *X)
{     //*** XFoil SCRIPT DEFINITION ***
   ofstream out("XFoil.txt");

   if(!out)
   {
      cout << "XFoil.txt error!!!" << endl;
      return 0;
   }

   out << "NACA 23012" << endl;

   out << "GDES" << endl;
   out << "TSET" << endl;
   out << X[0] << endl;
   out << X[1] << endl;
   out << "HIGH" << endl;
   out << X[2] << endl;
   out << X[3] << endl;
   out << "CADD" << endl << endl << endl << endl;
   out << "EXEC" << endl << endl;

   out << "PANE" << endl;

   out << "MDES" << endl;
   out << "FILT" << endl;
   out << "EXEC" << endl << endl;
```

```
    out << "OPER" << endl;

    out << "VPAR" << endl;
    out << "N " << 9 << endl;
    out << "VACC " << 0.001 << endl << endl;

    out << "a " << 3 << endl;
    out << "VISC " << 200000 << endl;
    out << "ITER" << endl;
    out << 40 << endl;
    out << "a " << 3 << endl;
    out << "INIT" << endl;
    out << "a " << 3 << endl;
    out << "PACC" << endl;
    out << "XfoilData.dat" << endl << endl;
    out << "a " << 3 << endl;
    out << "PACC" << endl << endl;

    out << "QUIT" << endl;

    out.close();

    // *** XFoil COMPUTATION ***

    system("@if exist XfoilData.dat del /q XfoilData.dat\n");
    system("call XFoil < Xfoil.txt > XfoilLog.txt");

    // *** DATA FROM Xfoil ***

    double a, CL, CD, CDp, CM, Top_Xtr, Bot_Xtr;

    XfoilData(a, CL, CD, CDp, CM, Top_Xtr, Bot_Xtr);

    //*** OBJECTIV FUNCTION ***

    return CD/CL;
}
```

This excerpt will be modified when solving the problems defined in the exercises. Symbols used in this script are consistent with the symbols used by XFoil (if necessary, check the guidebook). All symbols of the aerodynamic coefficients used in the script **are defined with lowercase letters, e.g. cl or cd** (according to the nomenclature used in the XFoil software). Aerodynamic coefficients used as variables in the OptoFoil software are defined in capital letters. At the end of the presented code, there is a definition of the objective function (e.g. CD/CL). It is a variable that is returned as the result of the XFoil function.

3. Post.cmd

The results obtained using OptoFoil contain the values of four decision variables which describe the geometry of the airfoil. To create the airfoil geometry, run the Post.cmd script (the script is included in the lab materials). Then follow the instructions. The software will ask you to enter the values of the decision variables obtained during the calculations. Based on this data, a airfoil geometry file will be created. This file should be exported to XFoil. On its basis, you can get acquainted with the geometry of the optimised airfoil and perform additional calculations, e.g. for angles of attack other than those used during optimisation.

Description of the most important parts of the code:

Row	Description.
66 – 69	Definition of initial values.
129	Xfoil function-the function that creates the script.
141 - 177	Preparation of the Xfoil script.
141	Name of the optimised airfoil.
194	Objective function definition e.g.. CD/CL.
325	The function that defines the stop criterion.

4. Exercise No.1

Task:

Optimise the aerodynamic efficiency of NACA 23012 airfoil

Data:

OptoFoil.cpp file.

Instructions:

Open the "OptoFoil.cpp" file. OptoFoil.cpp. Dev-cpp environment will be opened. The file contains the "main" function, optimisation algorithms and the objective function algorithm. Settings for this task should not need any modifications. OptoFoil should be simply compiled and run.

Analysed airfoil NACA23012
Vector of design variables X = {0.12; 0.0184; 0.3; 0.149}
AoA α = 3deg
Objective function maximise the aerodynamic efficiency

Code modifications:

No modifications are required. The script is ready to be run. Please note the definition of the objective function. Due to the applied gradient method, the objective function adopted for optimisation is inversed. i.e. when we are seeking minimum inversed efficiency , we get maximum efficiency.

The code should be compiled (compilation results are in the "OptoFoil.exe" program), and then program should be run. During the optimisation process, "Log.txt" files are generated with a record of the optimization process, "Results.txt" with optimisation results after each iteration, and "Xfoil.txt" with an automatically generated script, XfoilLog.txt with the Xfoil logo. To follow the progress of optimisation, open the "Log.txt" and "Results.txt" files.

Results:f

The analysis of the obtained results should include:

- Comparison between the airfoil geometry before and after the optimisation.
- Description of the improvement in efficiency with regards to changing the lift and drag coefficients.
- Calculations of airfoil characteristics using Xfoil and a graph which would illustrate them.
- Conclusions

5. Exercise No. 2

Task:

Minimise the CD and CM coefficients for the NACA 2412 airfoil at a constant CL=0.6

Data:

OptoFoil.cpp

Instructions:

1. Define the objective function and type it in the correct row in the source code.
2. Apply any necessary changes to the code.
3. Compile and run the code.

Results:

The analysis of the obtained results should include:

- Comparison between the airfoil geometry before and after the optimisation.
- Description of the changes to CD and CM coefficients.
- Conclusions

6. Exercise No. 3

Task:

Maximise the CL coefficient and minimise the CM coefficient for NACA 2412 at a constant AoA= 3 deg

Data:

OptoFoil.cpp

Instructions:

1. Define the objective function and type it in the correct row in the source code.
2. Apply any necessary changes to the code.
3. Compile and run the code.

Results:

The analysis of the obtained results should include:

- Comparison between the airfoil geometry before and after the optimisation.
- Description of the changes to CD and CM coefficients with regards to the objective function minimalisation.
- Conclusions (How much did the CL and CM coefficients change? Is it possible to regulate these changes: is it more important to get high lift coefficients or minimise CM? Could you propose any solutions? How does the pressure distribution look on the optimised airfoil ?-present it in a diagram).

7. Exercise No. 4

Task:

Minimise the CD coefficient for NACA 0012 at a constant CL=0.1.

Data:

OptoFoil.cpp

Instructions:

1. Define the objective function and type it in the correct row in the source code.
2. Apply any necessary changes to the code.
3. Compile and run the code.
4. Pay attention the the thickness of the airfoil.
5. Define the penalty function
6. Re-compile and re-run the code.

Results:

The analysis of the obtained results should include:

- Comparison between the airfoil geometry before and after the optimisation.
- Description of the changes to CD coefficients.
- Conclusions.