

Solarian Programmer

My programming ramblings



Home

Archives

Contact

Privacy

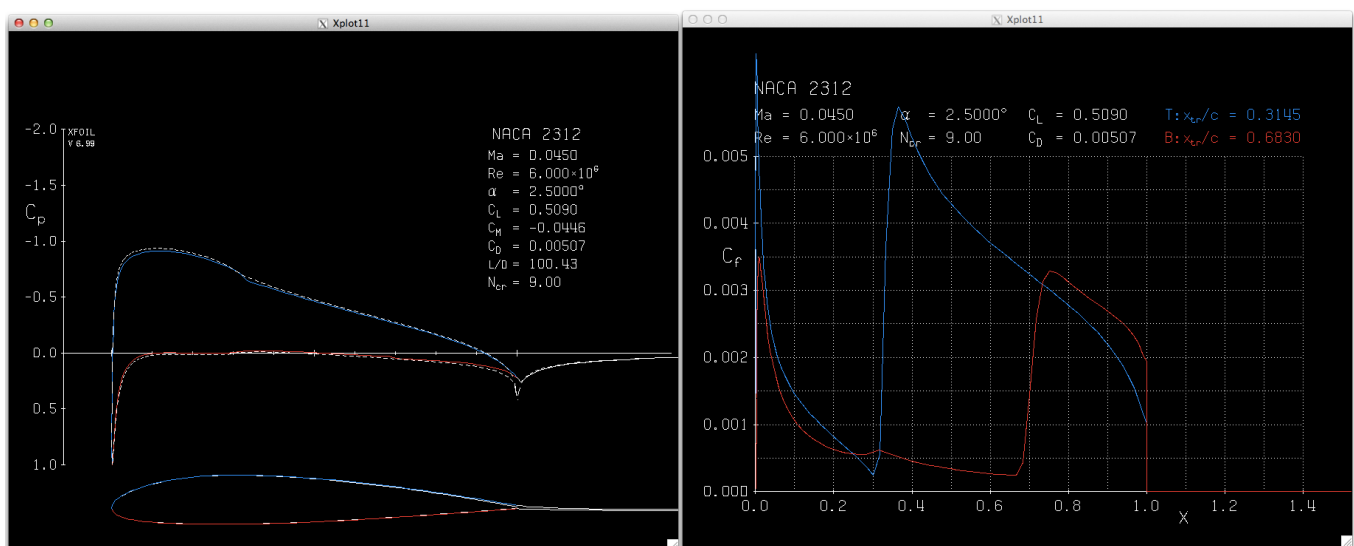
Compiling Xfoil on OS X

Posted on January 23, 2014 by Paul

If you are into airfoil analysis as an amateur or as a professional you will need a quick way to estimate the aerodynamic characteristics (lift, drag and momentum coefficients) of a given airfoil. The classical approach to calculate the performance of an airfoil is to use a coupled inviscid/viscous flow solver. On a modern computer this analysis will usually take less than a minute. Compare this with a full [CFD](#) approach that uses the [Navier-Stokes](#) equations for the laminar part of the flow and the [Reynolds Averaged Navier-Stokes](#) equations for the turbulent part of the flow. In the former case a typical analysis can run from a few minutes to a few hours on a modern PC.

[Xfoil](#) is an open source *interactive program for the design and analysis of subsonic isolated airfoils*. Xfoil uses a potential flow solver coupled with a [Boundary Layer](#) solver to calculate the flow field around an airfoil. The author, Mark Drela, provides a Windows binary on Xfoil's web page. If you are a Linux user building Xfoil from sources is straightforward. For OS X, well, it is a bit more complicated to build the code from sources.

A typical Xfoil result is presented in the next figure:



In order to be able to build and run Xfoil on OS X you will need an X server like [XQuartz](#). Xfoil is implemented in Fortran and C, so we'll need a C and Fortran compiler. [Xcode](#) will give you the C compiler. A Fortran compiler can be [built](#) from sources or installed from

Homebrew. On my machine, I've built **GCC** from sources with the C, C++ and Fortran compilers, but you can safely use the ones from Homebrew if you prefer.

Assuming that you have all the requirements (XQuartz, gfortran, gcc) installed, we can start the actual compilation of Xfoil. You can download the latest sources from Xfoil's web [page](#). In the remaining of this post I will suppose that you have extracted Xfoil in your home folder.

Open a Terminal and navigate to the *orrs* folder from Xfoil:

```
1 cd ~
2 cd Xfoil
3 cd orrs
```

Next we will need the absolute path to the *orrs* folder, this can be found with:

```
1 pwd
```

On my machine the result of the above command was */Users/sol/Xfoil/orrs*, be sure to use your own user name instead of **sol**.

Go further in the directory tree of Xfoil with:

```
1 cd src
```

and open *osmap.f* in your favorite text editor. At about line 101 you will find this line:

```
1 DATA OSFILE / '/home/codes/orrs/osmapDP.dat' /
```

change it to reflect the actual path to your *orrs* folder (the result of *pwd* from above), again be sure to use your own user name instead of **sol**:

```
1 DATA OSFILE / '/Users/sol/Xfoil/orrs/osmapDP.dat' /
```

Save and close *osmap.f*.

Next, we need to go in the *bin* folder of *orrs*:

```
1 cd ../bin
```

Open *Makefile* and replace all occurrences of */home/codes/Xplot* with */Users/sol/Xfoil/*. Find the compiler flags:

```
1 FC = f77
2 FLG = -O
3 PLTLIB = -lX11
4 FTNLIB =
```

and change them to:

```
1 FC = gfortran
2 FLG = -O2 -fdefault-real-8
3 PLTLIB = -lX11
4 FTNLIB =
```

Find the flags for the Intel Fortran Compiler:

```
1 FC = ifort
2 FLG = -O -fpe0 -CB
3 PLTLIB = -L/usr/X11R6/lib -lX11
4 FTNLIB =
```

and comment them:

```
1 #FC = ifort
2 #FLG = -O -fpe0 -CB
3 #PLTLIB = -L/usr/X11R6/lib -lX11
4 #FTNLIB =
```

Save the *Makefile* and write the next set of commands in your Terminal:

```
1 make osgen
2 make osmap.o
3 cd ..
4 bin/osgen osmaps_ns.lst
```

Next, we are going to build the plot library used by Xfoil:

```
1 cd ~/Xfoil
2 cd plotlib
```

Open the *Makefile* from *~/Xfoil/plotlib* and change:

```
1 INSTALLDIR = .
```

to

```
1 INSTALLDIR = /Users/sol/Xfoil/
```

Save and close the *Makefile*.

Open *config.make* and change

```
1 PLTLIB = libPlt_gSP.a
```

to

```
1 PLTLIB = libPlt.a
```

and:

```
1 #DP = -fdefault-real-8
```

to

```
1 DP = -fdefault-real-8
```

Save and close the file.

Next, run:

```
1 make libPlt.a
2 cd ~/Xfoil/bin
```

Edit the *Makefile* from *~/Xfoil/bin* by changing:

```
1 BINDIR = /home/codes/bin/
```

to

```
1 BINDIR = /Users/sol/Xfoil/bin/
```

Find the default compilers options:

```
1 FC = f77
2 FFLAGS = -O
3 FFLOPT = -O
4 INSTALLCMD = install -s
5
6 CC = cc
7 CFLAGS = -O -DUNDERSORE
```

and change them to:

```
1 #FC = f77
2 #FFLAGS = -O
3 #FFLOPT = -O
4 #INSTALLCMD = install -s
5
6 CC = gcc
7 CFLAGS = -O -DUNDERSORE
```

Find:

```
1 FC = gfortran
2 CHK =
3 CHK = -fbounds-check -finit-real=inf -ffpe-trap=invalid,zero
4 DBL = -fdefault-real-8
5 FFLAGS = -O2 $(CHK) $(DBL)
6 FFLOPT = -O2 $(CHK) $(DBL)
7 FTNLIB =
```

and add the next line at the end of the above lines:

```
1 INSTALLCMD = install -s
```

Also, change:

```
1 PLTOBJ = ../plotlib/libPlt_gDP.a
```

to

```
1 PLTOBJ = ../plotlib/libPlt.a
```

Save the *Makefile*.

Last steps in the compilation:

```
1 make xfoil
2 make pplot
3 make pxplot
```

Now, you should be able to run Xfoil with:

```
1 cd ~
2 cd Xfoil/bin
3 ./xfoil
```

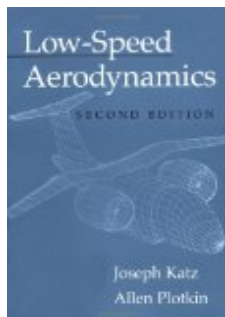
or, optionally, you can add the next line at the end of your *.bash_profile* file:

```
1 export PATH=/Users/sol/Xfoil/bin:$PATH
```

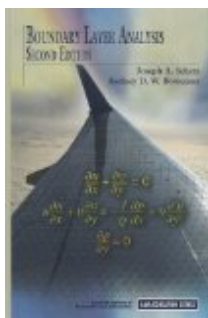
Save and run in the Terminal:

```
1 cd ~
2 . .bash_profile
3 xfoil
```

If you are interested in learning more about the method used by Xfoil for solving the inviscid flow field I would recommend reading [Low-Speed Aerodynamics](#) by J. Katz and A. Plotkin:



For a good introduction into the Boundary Layer theory and numerical solutions of these equations, a good introduction is [Boundary Layer Analysis](#) by J.A. Schetz and R.D.W. Bowersox:



Show Comments

[C#](#) [Charts](#) [C++11](#) [Regex](#) [Scheme](#) [Multithreading](#) [Posix](#) [Books](#) [C++](#) [C++14](#)
[C++17](#) [OSX](#) [Python](#) [Objective-C](#) [Windows](#) [Clang](#) [Fortran](#) [CUDA](#) [Roguelike](#) [Perl](#)
[Cling](#) [C++20](#) [Linux](#) [WSL](#) [Fractals](#) [OpenGL](#) [JavaScript](#) [OpenCV](#) [BeagleBone](#)
[Productivity](#) [Raspberry Pi](#) [OpenMP](#) [iOS](#) [Node.js](#) [macOS](#) [NumPy](#) [SciPy](#) [Matplotlib](#)
[GCC](#) [Swift](#) [C](#) [C99](#) [C11](#) [Arduino](#) [Videos](#) [Armadillo](#) [Chromebook](#) [ChromeOS](#)
[Docker](#) [x86-64](#) [F#](#) [dotnet](#) [C17](#) [Lisp](#)

Recent posts

- [Compiling GCC 9 on macOS Catalina](#)
- [Install OpenCV 4 on Raspberry Pi for C++ and Python development](#)
- [Building SBCL - Steel Bank Common Lisp on Windows](#)
- [Cross compiling OpenCV 4 for Raspberry Pi Zero](#)
- [Python - using C and C++ libraries with ctypes](#)
- [Install GNU Octave on macOS and getting started with the image processing package](#)
- [Batch convert images to PDF with Python by using Pillow or img2pdf](#)
- [C Programming - Reading and writing images with the stb_image libraries](#)
- [Building the Windows Terminal application with Visual Studio 2019](#)
- [C++17 STL Parallel Algorithms - with GCC 9.1 and Intel TBB on Linux and macOS](#)

Disclaimer:

All data and information provided on this site is for informational purposes only. solarianprogrammer.com makes no representations as to accuracy, completeness, currentness, suitability, or validity of any information on this site and will not be liable for any errors, omissions, or delays in this information or any losses, injuries, or damages arising from its display or use. All information is provided on an as-is basis. solarianprogrammer.com does not collect any personal information about its visitors except that which they provide voluntarily when leaving comments. This information will never be disclosed to any third party for any purpose. Some of the links contained within this site have my referral id, which provides me with a small commission for each sale. Thank you for understanding.

Copyright © 2019 - Paul Silisteanu