AVL Tutorial: Introduction, Installation, and Basic Usage

Prerequisites

The only prerequisite is to be able to install AVL. Access to any of the three main OSs available - Linux, Mac, or Windows - is sufficient.

What is AVL? How does it work? Why do we use it?

Before jumping into installation procedures, let's talk about AVL: what even is it?

AVL stands for Athena Vortex Lattice. It is a program developed by some people over at MIT and it is used to gather relevant aerodynamic figures - stability derivatives, lift distributions, etc. - for a given aircraft configuration.

In very practical terms, it is something you install that runs on your computer's command prompt (or terminal). There is no GUI involved. You input some files that define the geometry of your aircraft and its mass properties, as well as different flight conditions, and the program spits out a variety of different values, plots, and properties concerning the aerodynamics of your design.

It is thus easy to see why we use it: before choosing a particular configuration or geometry for our design, a series of analyses are run to make sure that the plane can fly, i.e., it is stable, controllable, and efficient (or at least somewhat stable, quasi-controllable, and borderline efficient). AVL helps us to determine whether these are true, and also allows us to make modifications on-the-fly to approach the best design possible.

This is all you need to know. Something about the name, though…

A note on vortex lattices: what the hell is a vortex and what in the world is a lattice. AVL's design space.

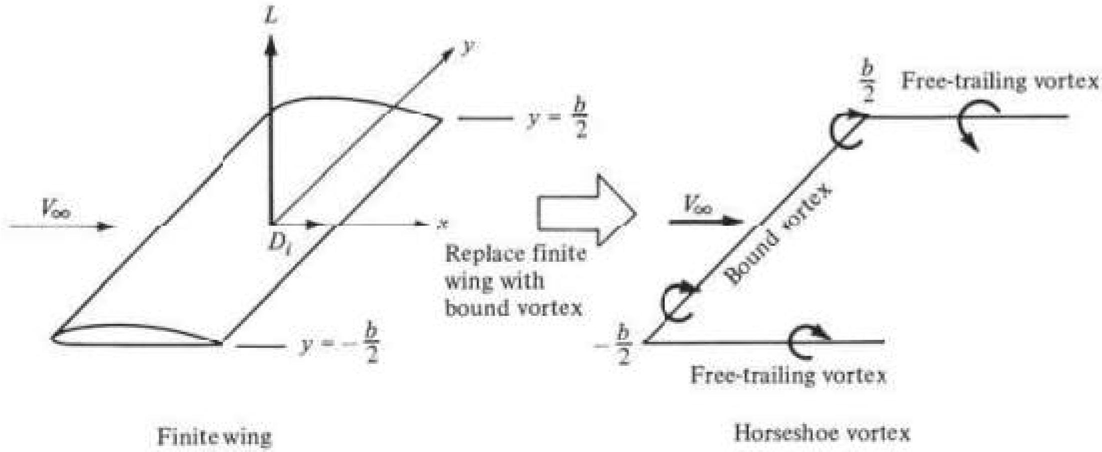
Imagine a wing.

A wing generates lift. This lift is not constant in the span-wise axis (from now on conveniently referred to as the y-axis). In fact, there exists a lift distribution over the wing, meaning that for each chord (think a cross-section of the wing, an airfoil if you will) along our span we can find the lift force that acts there. Finding this lift distribution is a big problem in CFD, and many physical models exist to numerically compute this distribution.

One family of such models is known as LVM, or lattice-vortex models. This comes from a wider set of theory called Lifting-Line Theory developed by Ludwig Prandtl. This is a huge name in Aerodynamics. It was Prandtl who first described boundary layers and their relationship with aerodynamic stalls. Lifting-Line Theory was also revolutionary, and is the basis of a lot of aerodynamic theory. Mandatory historical note: Prandtl did have some questionable affiliations and affirmations during WW2. Take that as you will.

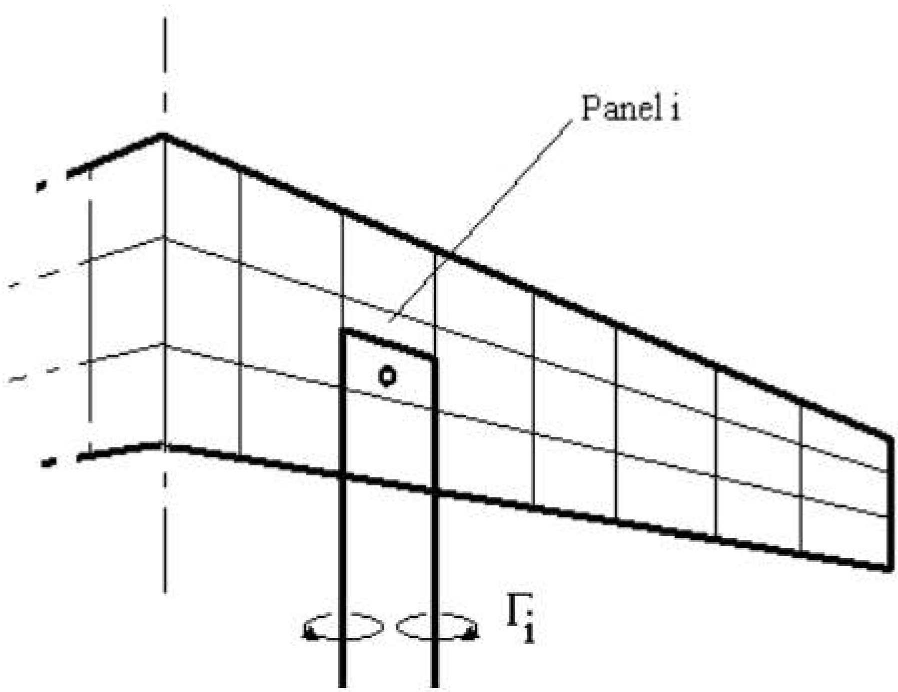
It is not crucial to know anything at all about LVM, but here is a *very* crude summary for the curious. A much more in-depth explanation can be found in John D. Anderson's *Fundamentals of Aerodynamics*, chapter 5 section 3. Great read.

Vortex-lattice models define wing planforms as a collection - or a lattice, if you will - of horseshoe vortices. Horseshoe vortices are composed of a vortex bound to the lifting surface and two trailing vortices, as shown below:



Placing an infinite number of these horseshoe vortices over a wing allow us to find the lift distribution. The reason why this is needed is explained in detail in Anderson, but a somewhat intuitive motivation to why a single horseshoe vortex doesn't suffice when it comes to modelling the lift distribution is the fact that the downwash in the trailing vortices would be infinite in magnitude (again, see Anderson).

In the case of AVL and the vortex-lattice method, the wing is modeled as a collection of panels, each panel containing a horseshoe vortex as shown below:



This is a basic summary of the operating principle behind AVL. There is obviously much more to this, and some good resources for further reading can be found in the end of this tutorial.

Now, to the most important part of all of this: due to the nature of the vortex-lattice model, this model works well for **thin lifting surfaces at small angles of attack and sideslip**. These trailing vortices are assumed to be parallel to the x-axis (downstream), hence small sideslip.

This design space is exactly where RC planes fall, and roughly where DBF competition planes fall. AVL uses a model sufficiently accurate for our purposes and flight conditions, but I don't think the new B-21 was designed using AVL.

Awesome, enough of this, let's get to the installation!

AVL installation and a note on directory and versions

The installation process is pretty simple. To install AVL, access the following website:

https://web.mit.edu/drela/Public/web/avl/

Scroll down until you see "Software:"

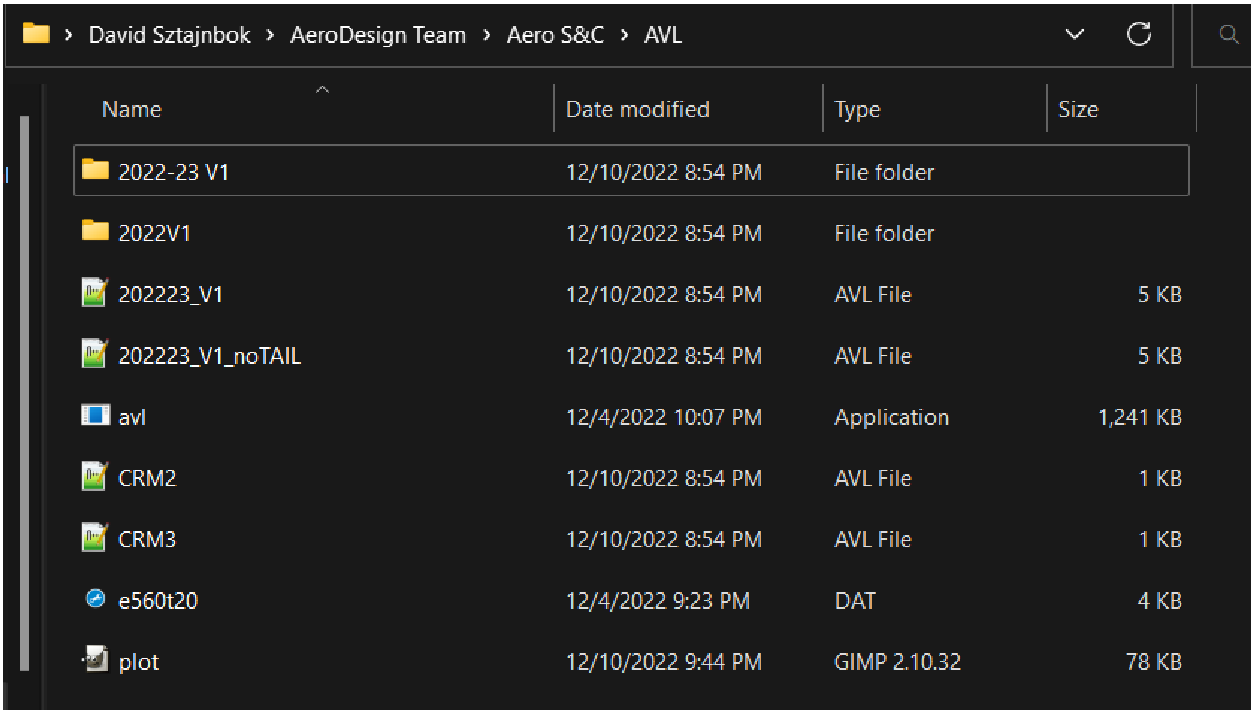
Software

Latest Versions

- avl3.40b.tgz (606752 bytes)
  AVL 3.40b for Unix and Win32. Gzipped directory tar image.
  All source code, User Guide, sample AVL session inputs.
  Requires Fortran, C compilers, windowing support.
- avl (1853240 bytes)
  AVL 3.40b executable for Intel x86_64. Requires xQuartz for graphics.
- avl (1685504 bytes)
  AVL 3.40b executable for ARM MacOS. Requires xQuartz for graphics.
- avl (1054104 bytes)
  AVL 3.40b executable for Linux 64. Requires libX11 libraries for graphics.
- avl.exe (1674868 bytes)
  AVL 3.40b executable for Windows.
- avl3.36.tgz (606752 bytes)
- avl3.36.exe.zip (509063 bytes)
  AVL 3.36 executable for Windows.
- avl3.35.tgz (1041053 bytes)
- avl335.zip (505104 bytes)
  AVL 3.35 executable for Windows.
- avl335_MacOSX.zip (95232 bytes)
  AVL 3.35 executable for Mac OSX. Requires Quartz library to be present on system.
- avl3.32.tgz (496506 bytes)
- avl332.zip (678020 bytes)
  AVL 3.32 executable for Windows.
- AVL3.27_MacOSX_10_4.zip (269506 bytes)
  AVL 3.27 executable for MacOSX.

Choose the best version for you. If you install the executable, create a folder for it. If you unzip one of the zip files, work in the directory that you extract the folder.

A note on directory: as will be explained later, AVL takes in a couple of files. Though in theory you can have these files whenever and reference their path in the terminal, this will sometimes not work for some reason, so it is best to just have these required files in the same directory as your AVL executable. This is what my directory looks like:

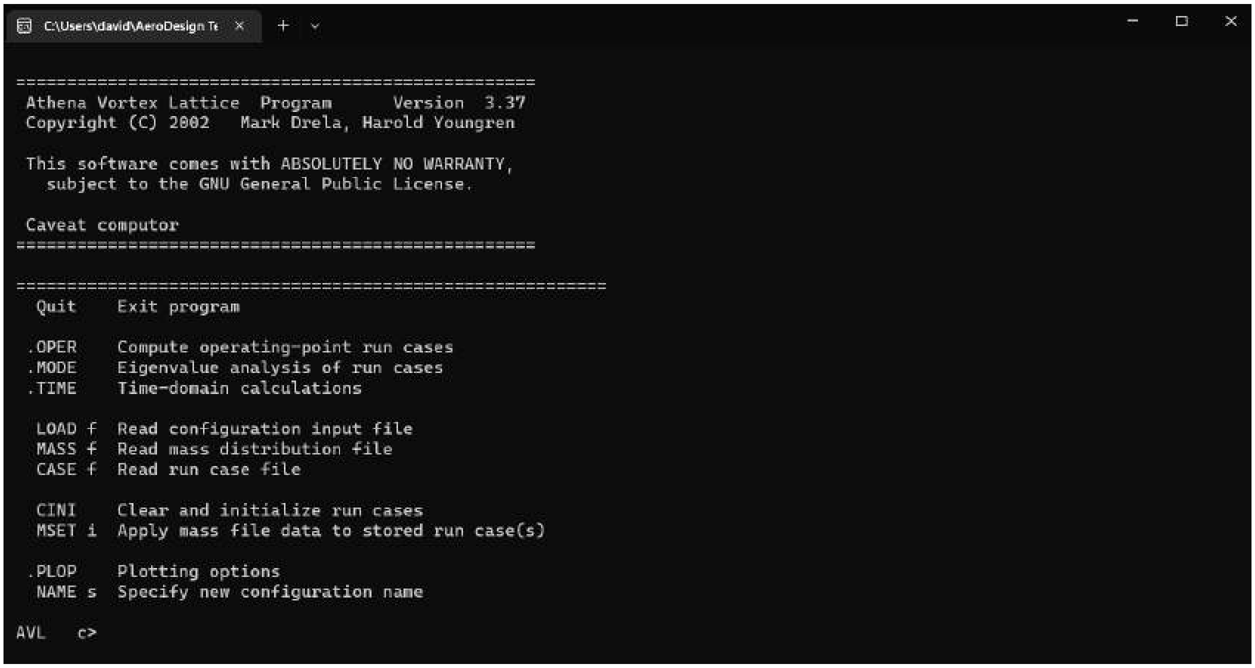


As you can see, not only is the AVL executable in there, but also other files required. As will be discussed later, “202223_V1” is a geometry file, while “CRM2” and “CRM3” are the run case files for missions 2 and 3, respectively, of the 2022-2023 DBF Competition. The other two files, “e560t20” and “plot” are, respectively, the airfoil used for V1 and an AVL output with a lift distribution plot as well as a Trefftz plot. More on these later.

Also, a note on versions: I have previously tried to use the latest version of AVL, 3.40, but for some reason some features were broken for me. I could not, for instance, rotate the aircraft geometry, and most visualization tools were broken. Using a previous version, 3.37, worked. Previous versions can be found a bit further down the page.

Basic usage: navigating the terminal

Once you have AVL installed, run the executable file, avl.exe. You should see a terminal pop up and be greeted with something like this:



This is AVL. It runs in your terminal, as was mentioned earlier.

The good thing about AVL is that basically everything you can do is shown in the screen right in front of you. All of the commands that you can run are displayed. In this case, for instance, we can run OPER, MODE, TIME, LOAD, MASS, etc.

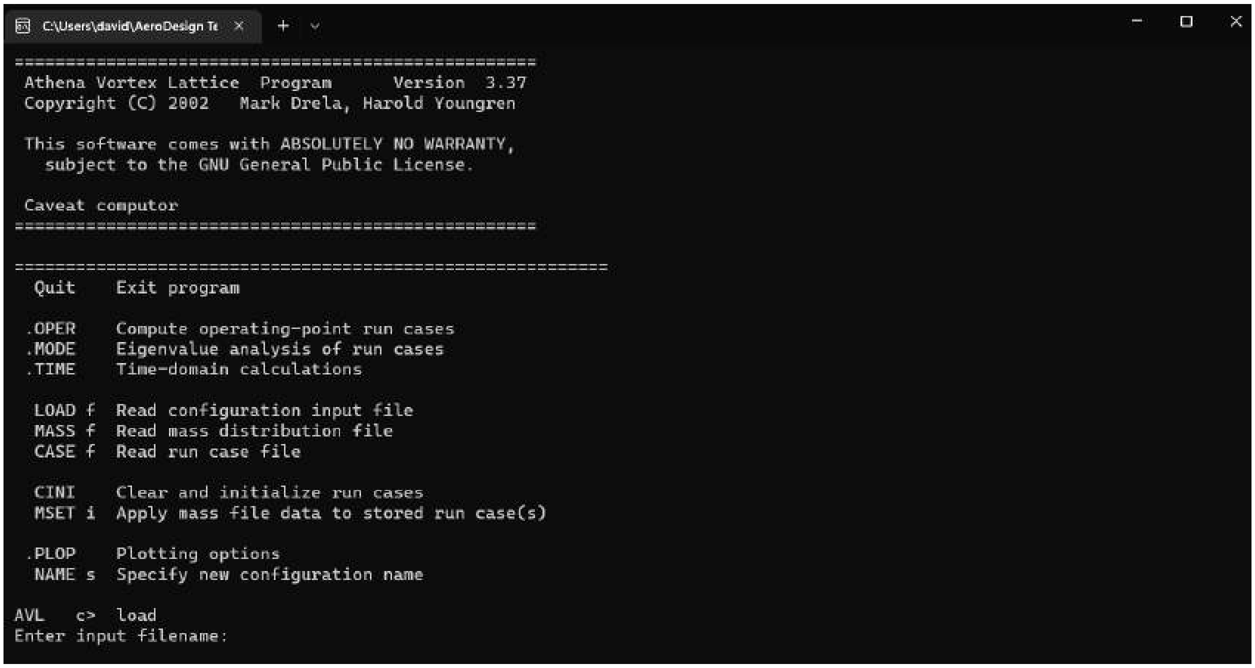
The first two are perhaps the most important commands. After we load our geometry and run case files, we will be ready to get those plots and figures we need for the aerodynamic analyses. The way we actually get these values is by entering either OPER or MODE. These are so-called **routines**, and they are the main ones. For the tutorials to follow, we will only cover **OPER**. This is the most relevant to us. OPER is short for operating menu, and is where AVL will run the flow analysis for the geometry we give under the run case conditions we give.

Basic usage: inputting files

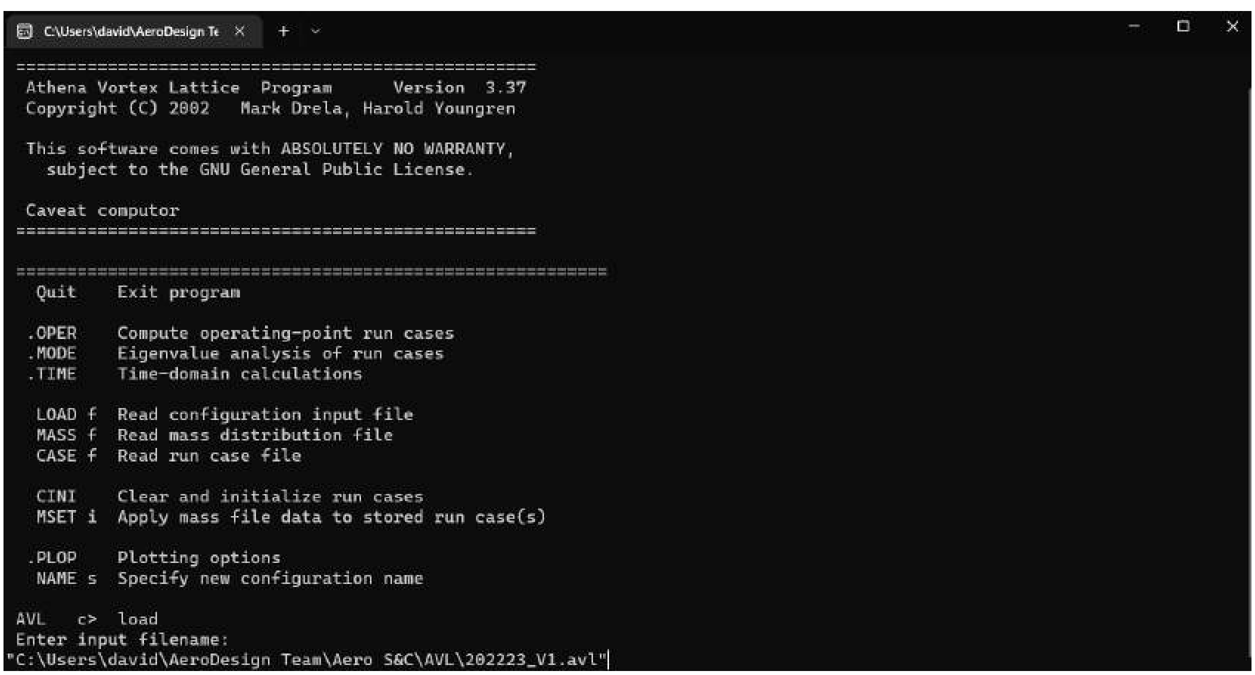
To wrap up this introductory tutorial, let's try inputting some files and getting into the OPER routine.

First, let's input our geometry file, that is, the file that defines the entire geometry of our aircraft. This is, of course, very different from a CAD model. We consider mainly the planform of lifting surfaces and a slender-body model of the fuselage. We will see what these look like in a little bit.

First, run the AVL executable. Then, type in “load” and hit enter. You should see something like this:



It is asking for the filename. Simply go to your file explorer, find the geometry file, and drag and drop it into the terminal. You should now see something like this:

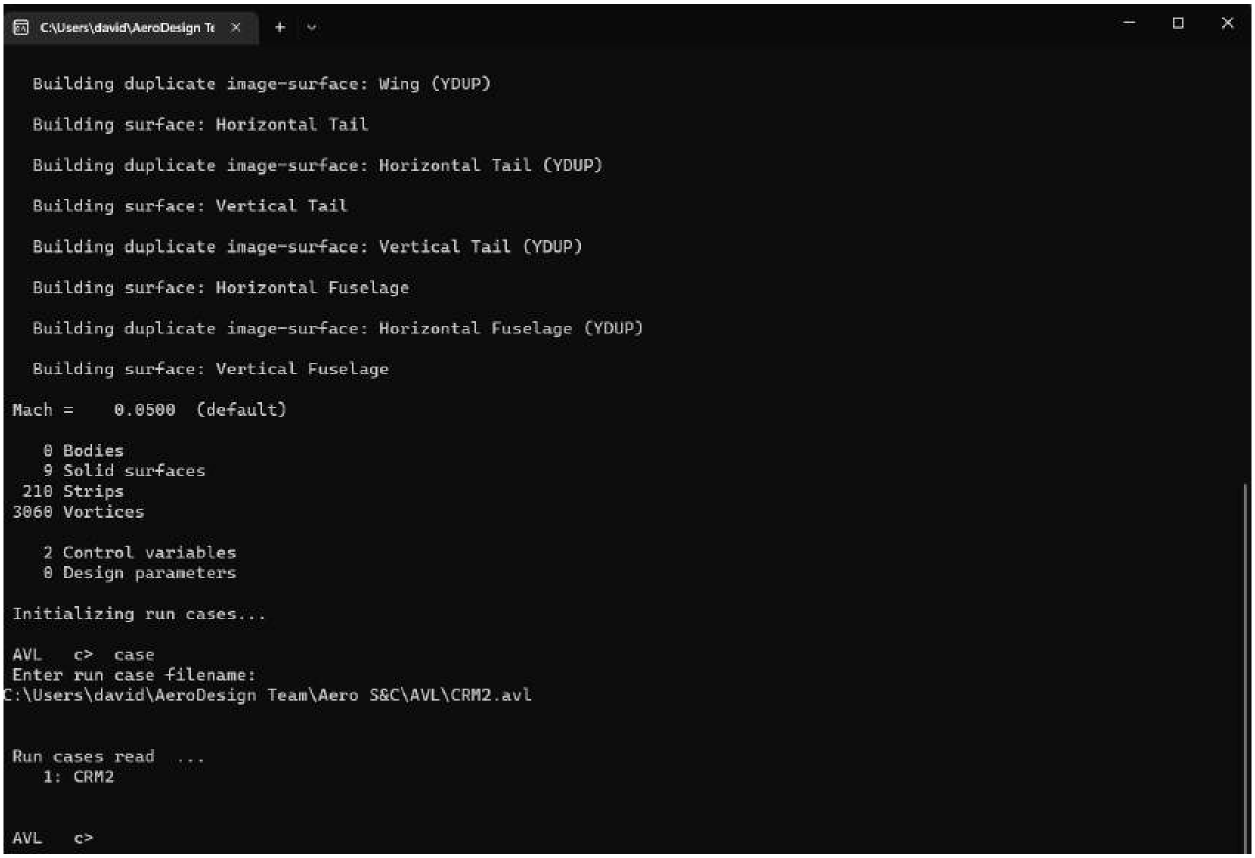


Notice that it added a set of quotation marks around the path. This doesn't happen to everybody, and it seems to depend on your OS and the terminal that you are using. If it does add these quotation marks, simply remove them. Hit enter and you should be taken back to the main menu:

```
AVL   c>  load
Enter input filename:
C:\Users\david\AeroDesign Team\Aero S&C\AVL\202223_V1.avl

Reading file: C:\Users\david\AeroDesign Team\Aero S&C\AVL\202223_V1.avl  ...

Configuration: competition_v1

  Building surface: Wing
    Reading airfoil from file: e560t20.dat
    Reading airfoil from file: e560t20.dat
    Reading airfoil from file: e560t20.dat

  Building duplicate image-surface: Wing (YDUP)

  Building surface: Horizontal Tail

  Building duplicate image-surface: Horizontal Tail (YDUP)

  Building surface: Vertical Tail

  Building duplicate image-surface: Vertical Tail (YDUP)

  Building surface: Horizontal Fuselage

  Building duplicate image-surface: Horizontal Fuselage (YDUP)

  Building surface: Vertical Fuselage

Mach =    0.0500  (default)

   0 Bodies
   9 Solid surfaces
 210 Strips
3060 Vortices

   2 Control variables
   0 Design parameters

Initializing run cases...

AVL   c>
```

Sweet! Time to add the run case files.

Repeat the exact same process, but instead of typing "load," type "case" as the command. Drag and drop the run case file (in my case, "CRM2" for mission 2), remove the quotation marks if needed, and hit enter. You should see this again:



Great! We have now loaded our aircraft's geometry and the run case conditions into AVL. Time to solve stuff.

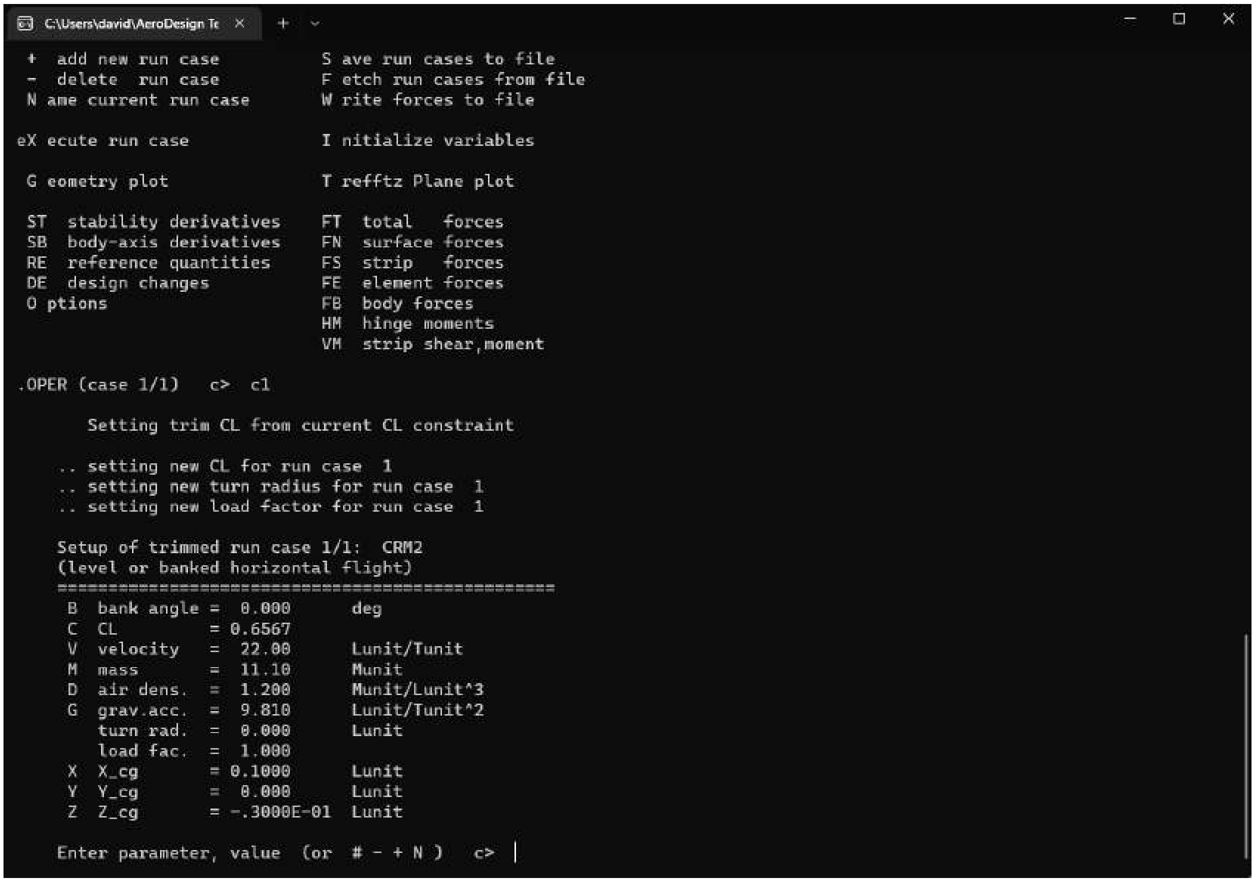
Basic usage: OPER routine

Let's enter the main operational routine, OPER. Type "oper" and hit enter:

```
C:\Users\david\AeroDesign Te
AVL   c>  oper

Operation of run case 1/1:   CRM2
==========================================================

 variable          constraint
 ------------      ------------------------
 A lpha        ->  CL          =   0.000
 B eta         ->  beta        =   0.000
 R oll  rate   ->  pb/2V       =   0.000
 P itch rate   ->  qc/2V       =   0.000
 Y aw   rate   ->  rb/2V       =   0.000
 D1  Flaps     ->  Flaps       =  -8.000
 D2  Elevator  ->  Cm pitchmom =   0.000
 ------------      ------------------------

 C1  set level or banked  horizontal flight constraints
 C2  set steady pitch rate (looping) flight constraints
 M odify parameters

"#" select  run case          L ist defined run cases
 +  add new run case          S ave run cases to file
 -  delete  run case          F etch run cases from file
 N ame current run case       W rite forces to file

eX ecute run case             I nitialize variables

 G eometry plot               T refftz Plane plot

 ST  stability derivatives    FT  total   forces
 SB  body-axis derivatives    FN  surface forces
 RE  reference quantities     FS  strip   forces
 DE  design changes           FE  element forces
 O ptions                     FB  body forces
                              HM  hinge moments
                              VM  strip shear,moment

.OPER (case 1/1)   c>
```

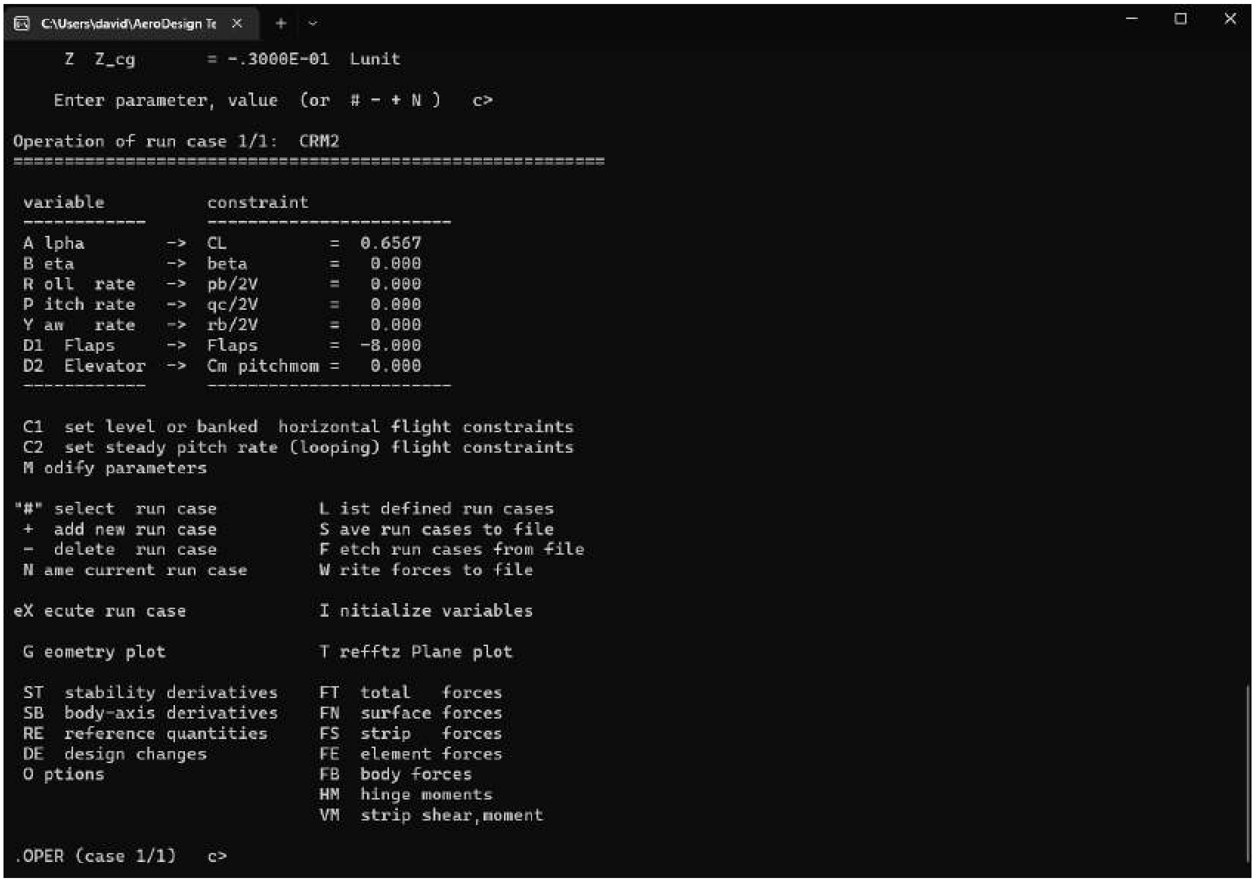
Notice that we have a lot of options now in terms of what we want to look at. Before investigating any of these further, we first need to actually run the program to find the numerical solutions under the run case that we input.

First, we have the option to select either C1 (horizontal flight) or C2 (looping flight). For the sake of simplicity, let’s go with horizontal flight. Type “c1” and press enter.



```
C:\Users\david\AeroDesign Te

 +  add new run case          S ave run cases to file
 -  delete  run case          F etch run cases from file
 N ame current run case       W rite forces to file

eX ecute run case             I nitialize variables

 G eometry plot               T refftz Plane plot

 ST  stability derivatives    FT  total   forces
 SB  body-axis derivatives    FN  surface forces
 RE  reference quantities     FS  strip   forces
 DE  design changes           FE  element forces
 O ptions                     FB  body forces
                              HM  hinge moments
                              VM  strip shear,moment

.OPER (case 1/1)   c>  c1

       Setting trim CL from current CL constraint

    .. setting new CL for run case  1
    .. setting new turn radius for run case  1
    .. setting new load factor for run case  1

    Setup of trimmed run case 1/1:  CRM2
    (level or banked horizontal flight)
    =================================================
     B  bank angle =  0.000      deg
     C  CL         = 0.6567
     V  velocity   =  22.00      Lunit/Tunit
     M  mass       =  11.10      Munit
     D  air dens.  =  1.200      Munit/Lunit^3
     G  grav.acc.  =  9.810      Lunit/Tunit^2
        turn rad.  =  0.000      Lunit
        load fac.  =  1.000
     X  X_cg       = 0.1000      Lunit
     Y  Y_cg       =  0.000      Lunit
     Z  Z_cg       = -.3000E-01  Lunit

    Enter parameter, value  (or  # - + N )   c>
```

For now, we will keep these parameters as they are. They come from the run case file, CRM2. Hit enter again to keep them as they are. You should be taken back to the "home screen:"



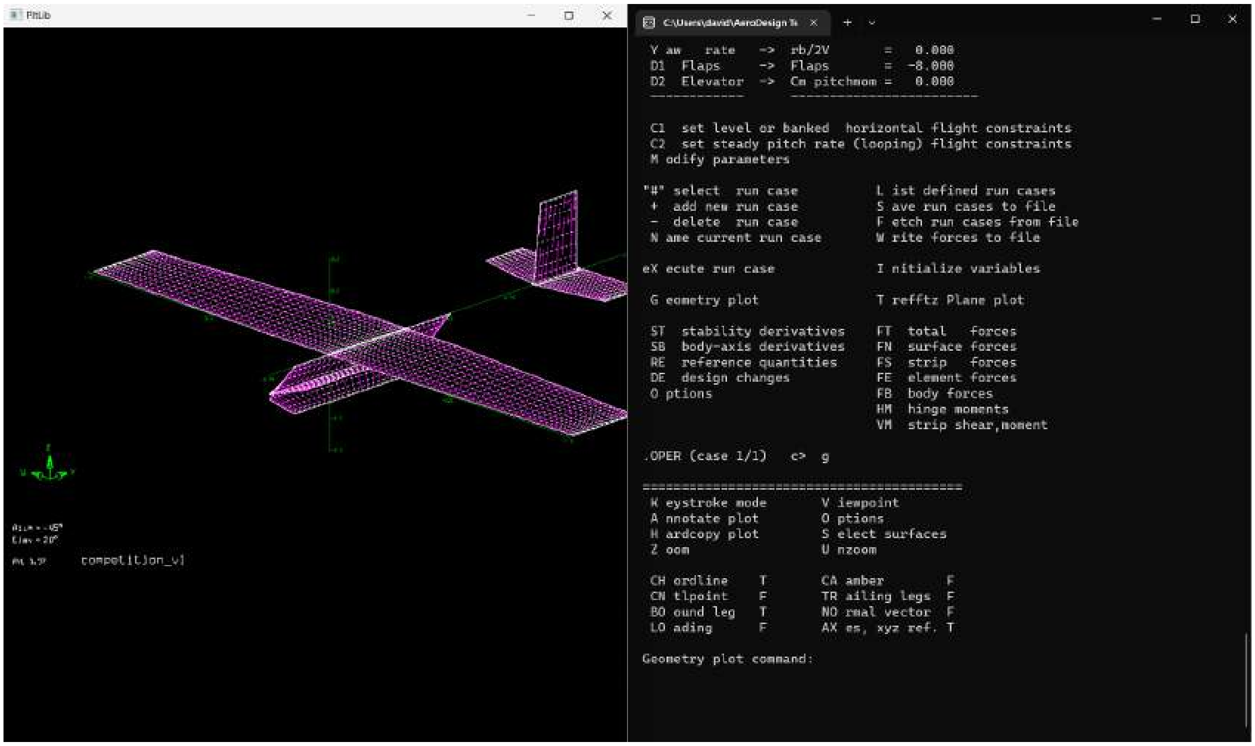
```
C:\Users\david\AeroDesign Te

     Z  Z_cg       = -.3000E-01  Lunit

    Enter parameter, value  (or  # - + N )   c>

Operation of run case 1/1:  CRM2
==========================================================

 variable          constraint
 ------------      ------------------------
 A lpha        ->  CL          =  0.6567
 B eta         ->  beta        =   0.000
 R oll  rate   ->  pb/2V       =   0.000
 P itch rate   ->  qc/2V       =   0.000
 Y aw   rate   ->  rb/2V       =   0.000
 D1  Flaps     ->  Flaps       =  -8.000
 D2  Elevator  ->  Cm pitchmom =   0.000
 ------------      ------------------------

 C1  set level or banked  horizontal flight constraints
 C2  set steady pitch rate (looping) flight constraints
 M odify parameters

"#" select  run case          L ist defined run cases
 +  add new run case          S ave run cases to file
 -  delete  run case          F etch run cases from file
 N ame current run case       W rite forces to file

eX ecute run case             I nitialize variables

 G eometry plot               T refftz Plane plot

 ST  stability derivatives    FT  total   forces
 SB  body-axis derivatives    FN  surface forces
 RE  reference quantities     FS  strip   forces
 DE  design changes           FE  element forces
 O ptions                     FB  body forces
                              HM  hinge moments
                              VM  strip shear,moment

.OPER (case 1/1)   c>
```

Perfect! All conditions are set, and AVL is ready to crunch some numbers. To execute the run case, type “x” and hit enter. This might take a little while. After it is done computing, it will output a pretty big verbose, but you should be back to the OPER menu:

```
CLff  =   0.65683     CDff  = 0.0238351     | Trefftz
CYff  =   0.00000         e =    0.8225     | Plane        Verbose up here

 Flaps          =  -8.00000
 Elevator       =   2.20636

---------------------------------------------------------------

Operation of run case 1/1:   CRM2
==========================================================

 variable          constraint
 ------------      ------------------------
 A lpha        ->  CL          =  0.6567
 B eta         ->  beta        =   0.000
 R oll  rate   ->  pb/2V       =   0.000
 P itch rate   ->  qc/2V       =   0.000
 Y aw   rate   ->  rb/2V       =   0.000
 D1  Flaps     ->  Flaps       =  -8.000
 D2  Elevator  ->  Cm pitchmom =   0.000
 ------------      ------------------------

 C1  set level or banked  horizontal flight constraints
 C2  set steady pitch rate (looping) flight constraints
 M odify parameters

"#" select  run case          L ist defined run cases
 +  add new run case          S ave run cases to file
 -  delete  run case          F etch run cases from file
 N ame current run case       W rite forces to file

eX ecute run case             I nitialize variables

 G eometry plot               T refftz Plane plot

 ST  stability derivatives    FT  total   forces
 SB  body-axis derivatives    FN  surface forces
 RE  reference quantities     FS  strip   forces
 DE  design changes           FE  element forces
 O ptions                     FB  body forces
                              HM  hinge moments
                              VM  strip shear,moment

.OPER (case 1/1)   c>
```

Now, you can look at whatever it is that you want to look. Some of these we will explore in greater detail. For now, to serve as a quick example, let's look at the geometry plot of our aircraft, or how AVL sees the plane that we input as the .avl file earlier. In other words, let's look at what this vortex-lattice is all about!

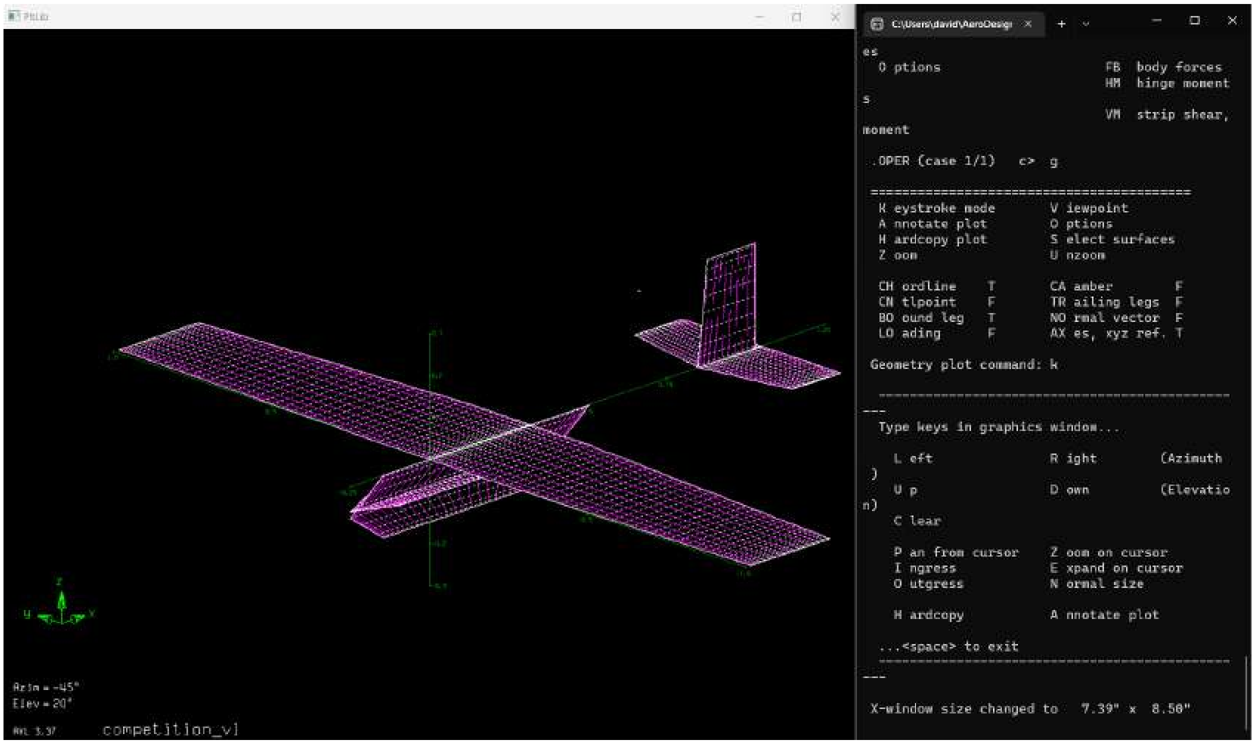
Type "g" and hit enter. A new window should open, showing you the aircraft model:



Notice some things. First, look at our fuselage! This is that slender-body model that I mentioned earlier. Assuming that the body is slender allows us to make some very convenient assumptions when it comes to the nature of the flow around it. See Anderson for more.

Also notice the lifting surfaces. They are quite thin, and that is not a mistake. Don't worry, AVL does take into consideration the airfoil selected. It is given as an input in the geometry file, which we will dive deeper into in a later tutorial. And look at that gap between the fuselage and the tail. That is where our tail boom will be!

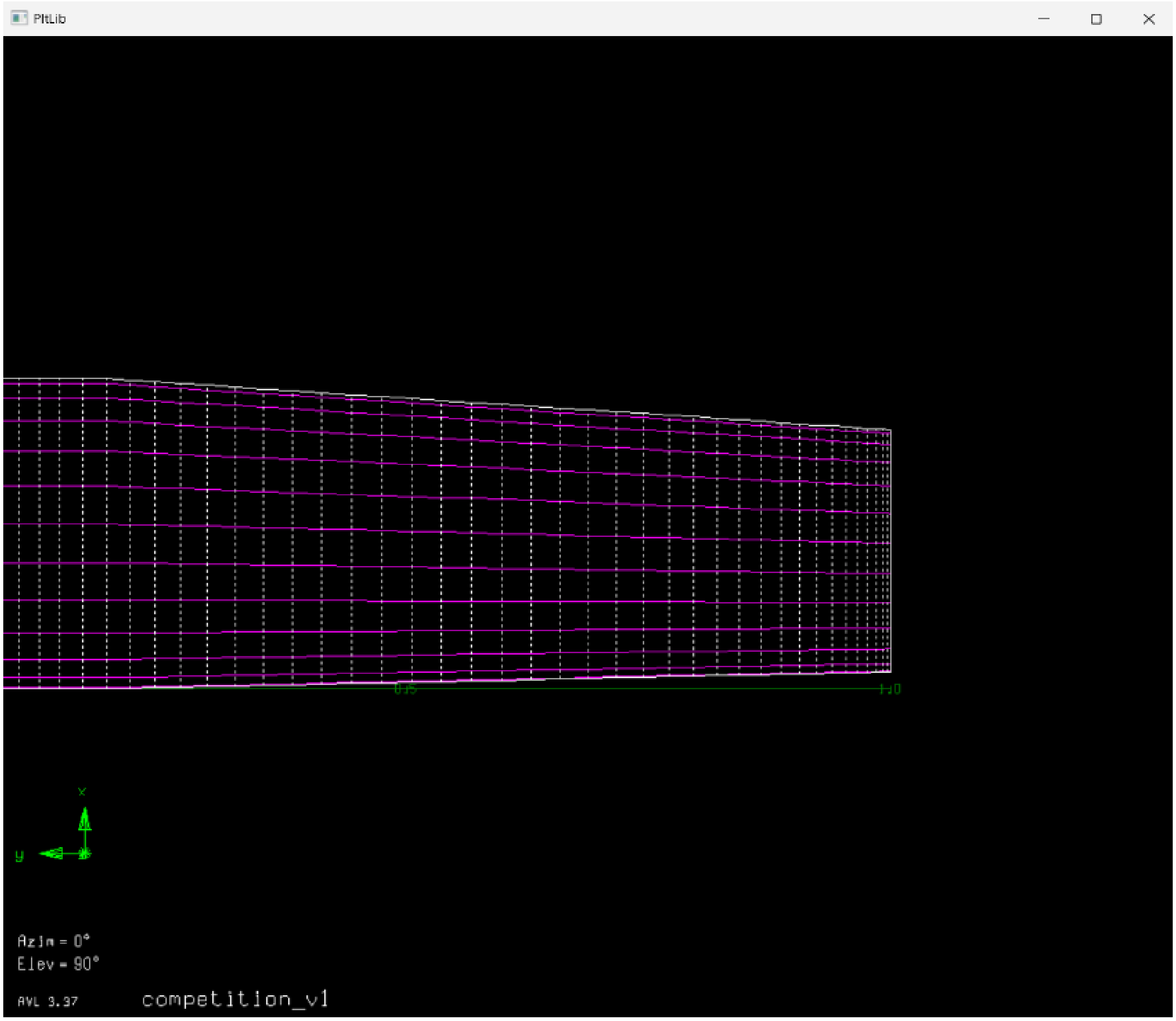
Let's manipulate this plot a bit. Notice how we have some options in AVL. If you want to move the model around, type "k" and hit enter. You are now in keystroke mode:



If you enter the geometry plot window by, say, clicking on it, you can hit the keys "L," "R," "U," and "D" to move the plot around. You can also pan around with "P," zoom with "Z," and do some other cool things as the AVL terminal shows. Play around with these for a bit, and get used to interpreting what each key means by what AVL tells you there.

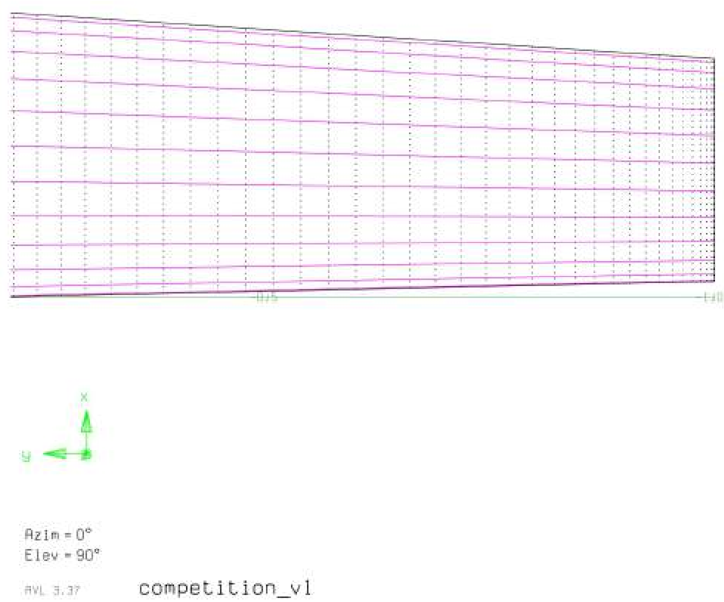
Note: this is where I got stuck when using AVL 3.40. I had to downgrade to 3.37 to be able to move the aircraft geometry plot around. If you are stuck in this step, doing what I did might be a good bet. Back to it!

If you zoom in on a lifting surface, you can even see some of the inner workings of AVL. This is the left section of the wing, near the tip:



Notice how clear the little pannels are where the horseshoe vortices are modelled. If you see this sort of tessellation in other aerodynamics-related software, it might have something to do with LVM.

Here again with a white background originally in PDF format, courtesy of AVL's own plotting function:



Final remarks and next steps

This is it for this introductory tutorial to AVL.

There is much beyond what was discussed here. Namely, the main gist of the work lies in creating the files we input to AVL - geometry and run case. One tutorial will be dedicated to each file, followed by more tutorials on other cool functions that we might use from time to time.

I hope you enjoyed and, as always, feel free to reach out with any questions. Happy AVLing!

Useful resources mentioned

Aerodynamics for Students' *3D Vortex Lattice Method*

John D. Anderson's *Fundamentals of Aerodynamics*

AVL's Website