Aerodynamic Analysis of F-16

Carter Tegen

December 14th, 2022

Contents

Executive Summary

This report analyzes the aerodynamic properties of the F-16 through different Mach numbers and angles of attack. Specifically, I used Pointwise, Star-CCM+, and Georgia Tech's Pheonix Cluster to conduct a full aerodynamic sweep and analyze how different mesh qualities affect the results at different Mach ranges. The results validate analytical aerodynamic models, displaying linear lift curve, simple drag polar model, drag divergence, as well as Prandtl-Glauerts compressibility correction. More generally, I created a framework that can easily analyze performance parameters through an entire flight envelope for a given vehicle.

Zipline Hiring Manager: This was a self-proposed project for my final project in my CFD class. It was done entirely by myself, with minimal help from peers in industry when I encountered a blocker. It's performing a full aerodynamic model of an F-16.

1 Introduction

1.1 Background

In the modeling and simulation of Aerospace vehicles, it's critical to understand the aerodynamic properties through the entire flight envelope. While wind tunnel tests are often the best option to gain accurate reports of this information, they are simply too costly and for projects that involve heavy iteration on vehicle designs. That's why in practice, computational solvers are often used as a cheap alternative, providing the ability to yield fast results of aerodynamic properties. The motivation of this project is to replicate some of the processes performed in industry that are used to analyze vehicles and create aerodynamic decks that can be fed into other tools.

More specifically, the aerodynamic properties of an aircraft, such as its lift, drag, and stability, play a major role in its ability to take off, fly, and land safely and efficiently. Aerodynamic analysis is used to understand and predict the aerodynamic behavior of an aircraft, and to optimize its design for maximum performance. By analyzing the aerodynamic properties of a proposed design, engineers can identify potential issues and make modifications to improve the performance of the aircraft. For example, if an aircraft has too much drag, it may not be able to achieve the required airspeed for takeoff, or it may have poor fuel efficiency during flight. Through aerodynamic analysis, engineers can identify areas of the aircraft that contribute to drag and make changes to reduce it.

1.2 Pointwise

Cadence Pointwise is the industry standard software for generating high fidelity meshes of geometries. Not only is it incredibly fast, but it also provides users unwavering control over the way meshes are generated. Pointwise has an ability to generate structured, unstructured, and hybrid meshes, allowing users to choose the best type of mesh for their specific problem. Users can easily repair surfaces, control mesh densities, and modify mesh generation algorithms. With this power comes quite the learning curve. While many solvers have integrated meshers that are straightforward and relativity forgiving, Pointwise can be quite the opposite. Because of this, it is best used in cases where an engineer has a strong understanding of what is desired out of a mesh.

Pointwise uses its own terminology to refer to parts of mesh and geometry. In short, a geometry is imported as a *Database*, which is composed of many *quilts* that stitch together to form a *model*. On the mesh side of things, connectors are formed as the borders of *quilts*, and have specified mesh points on them. *Domains* are the mesh way of representing a domain and are used to generate the surface mesh. Once a surface mesh is established, a volume can be assembled in a *block*

1.3 Star-CCM+

Star-CCM+ is a comprehensive computational fluid dynamics (CFD) simulation software package used in a variety of engineering fields, including aerospace, automotive, and maritime industries, to simulate the flow of fluids and the effects of the fluid on solid objects. The software allows users to simulate a wide range of physical phenomena, such as aerodynamics, heat transfer, and turbulence in both steady and unsteady flows. One of the key strengths of Star-CCM+ is its ability to handle large and complex models.

In addition to its power and ease of use, Star-CCM+ has an incredibly well documented Java API that allows you to control simulations through code. This makes it pretty straight-forward to write scripts to be executed by the PACE Pheonix cluster.

1.4 PACE Pheonix

The PACE Pheonix cluster is a high-performance computing (HPC) system that is used for academic and research applications. The cluster is composed of a large number of interconnected computer nodes, working together to provide powerful parallel computing. While modern computers are incredibly powerful, more expansive solution matrices and rigorous models require far more powerful computers to solve in an adequate amount of time. Hence, this project was solved using Georgia Tech's PACE Pheonix cluster.

2 Problem Formulation

2.1 Overview

Creating an aerodynamic deck for the F-16 requires us choosing different Mach numbers and angles of attack to analyze the flow. For Mach numbers, values were chosen up to Mach 2, the top speed of the F-16. Additionally, they were clustered around transonic, as this is where we can expect more dramatic changes in performance parameters. For angles of attack, values were clustered around 0, and went of up to 11 degrees. The scripts also allow sideslip to be tested, but this was ignored and simplified as it would dramatically increase the number of cases. As such, the following values were chosen:

$$Mach\ Array = \{0.1, 0.3, 0.7, 0.9, 0.95, 0.99, 1, 1.05, 1.3, 1.6, 2\}$$

$$AOA\ Array = \{-2, -1, 0, 1, 2, 3, 5, 7, 9, 11\}$$

These two arrays create a run matrix of 110 cases. While this is relatively coarse, the simplifying assumption of inviscid flow, which will be touched upon later, will generally lend itself to linear lift curves and parabolic drag polars.

2.2 Geometry

The geometry for this project was found on an online CFD forum[3]. A large reason why this specific geometry was chosen is because it was similarly used in a CFD project, making it likely that it would not contain many issues that would make meshing difficult.

2.3 Mesh

Mesh quality is not only a deterministic factor in the precision of CFD results, but also a major driver in the convergence of solutions. In order to see how mesh refinement affects the resulting performance parameters, the final aerodynamic decks were created on a coarse and fine mesh, tested both with and without Star's mesh adaptation. Additionally, a 5th viscous mesh was created to see the difference in drag from the inviscid and viscous cases.

2.3.1 Meshing Procedure

While I hoped that using a geometry that was used in another CFD project would make the meshing process relatively straight forward, this was anything but the case. By far the most difficult part of this process was learning Pointwise and the meshing process, which was all done through trial and error. However, I've learned how incredibly powerful Pointwise is compared to integrated meshers and I'm very grateful to have learned it.

Geometry Cleanup The first issue with the geometry that became apparent during the meshing process is that the model being used contains nearly a dozen incredibly tiny quilts between the cockpit and fuselage. These were deleted to have a cleaner mesh between the two objects and improve the area ratio of the surface mesh.

The second modification to the geometry was splitting the wing surface at the leading edge into two individual quilts, which assisted Pointwise in generating quads off the leading edge.

Surface Mesh In this project, I generated two different meshes, a coarse and fine meshes using average cell spacings of 0.1m and 0.06m respectively. Upon creating the domains on database entries, the first step was to merge and join connectors to clean up the geometry. This has immediate drastic effects on making the surface mesh appear more consistent and regular with cell sizing.

Even with the fine mesh, the initial domains struggled to capture areas of high curvature such as the nose cone, wing, and tail leading edges. For the wing and tail leading edge I applied 2d T-Rex, creating small cells that grow in size as they get further away from the leading edge as seen in Figure 1. Additionally, I increased cell density near the root and tips of all of the aerodynamic surfaces, as well as the nose tip. Figure 2 shows the sizing in these two meshes side by side with the same scale, where blue cells are small and red cells are large.

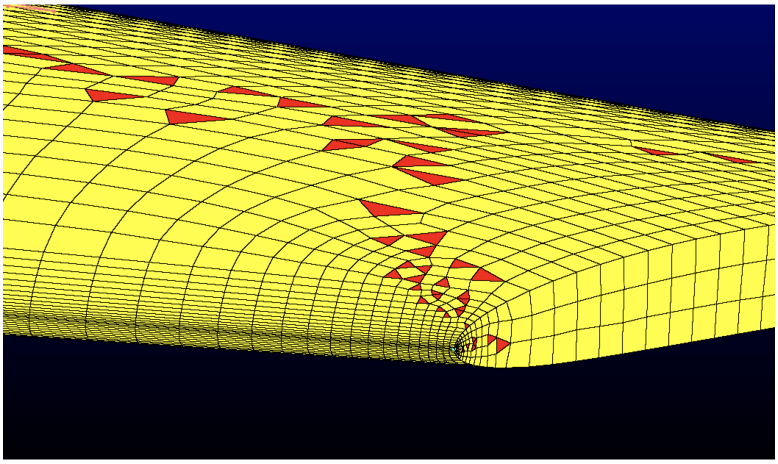
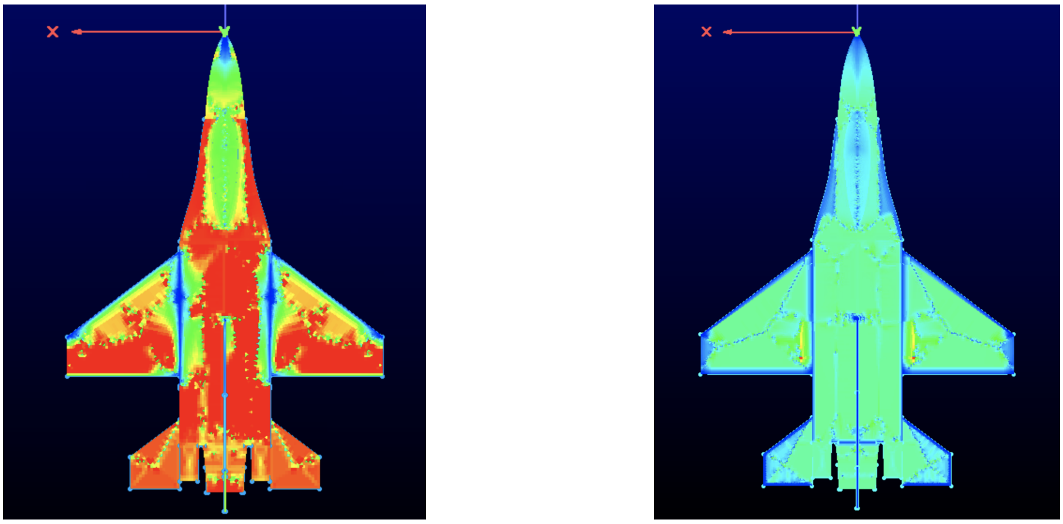


Figure 1: Cell growth on wing leading edge

Volume Mesh The volume mesh was established using a spherical farfield with a radius of 20 body lengths. It was initialized with no other modifications. The resulting sizes for coarse and fine meshes are 900k and 2.4M elements respectively.

2.3.2 Viscous Mesh

The viscous mesh was built upon the fine inviscid surface mesh, using 3D T-rex to generate the boundary layer. Using Cadence's y+ calculator[2], I used a specified initial spacing that would be appropriate for a K-epsilon model. An image of the boundary layer growth over the entire vehicle is seen below in Figure 3. Note that it may be difficult to view the growth on concave regions of the geometry, as growth in those regions ends sooner.



(a) Coarse Mesh (b) Fine Mesh

Figure 2: Mesh size comparison

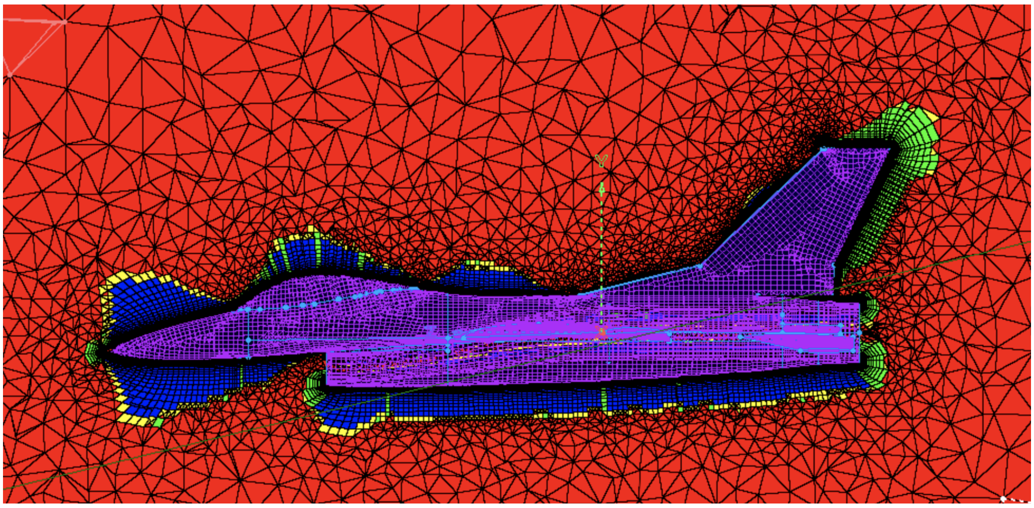


Figure 3: Anistropic boundary layer cells

2.4 Simulation

The simulations were executed by Generate_Deck.java and Submit_star.pbs, located in Appendix A and B respectively. All of the main simulations were inviscid and conducted at sea level. Additionally, the simulations were converged when continuity reached a threshold of 1E-3, and normal and axial force were both asymptotic for 40 iterations with a threshold of 0.005 for normalized difference between Min and Max. There was definitely room to improve on these convergence thresholds to ensure higher output precision, however, this gave some breathing room for convergence with lower quality meshes.

2.4.1 Generate Deck.java

The Generate_Deck script is used in conjugate with the Pheonix Cluster to instruct Star-CCM how to carry out all of the simulations. As previously mentioned, the main function of the script uses nested for loops to iterate through all of the freestream conditions. After all of the tests are conducted, the outputs are printed out to a CSV file. The other important constants are located below in Table 4.

Before running each case, the script updates the farfield boundary conditions and model initial conditions to match the mach number and flow direction. It does so by calling calculateFlowAngle, which uses the following equations to calculate the flow angle. Note, the coordinate system provided by the geometry has the positive z axis pointing forward and positive y upward.

Parameter	**Value**
Temp Ref	288.15 K
ρ Ref	1.225 kg/m^3
Pressure Ref	103,225 Pa
Surface Area (S) Ref	39.58 m^2
S Vertical Tail Ref	5.92 m^2
Speed of Sound Ref	340 m/s

Figure 4: Reference values in Generate Deck.java

$\hat{\mathbf{v}}_x = \sin(\beta)$
$\hat{\mathbf{v}}_y = \sin(\alpha) \cdot \cos(\beta)$
$\hat{\mathbf{v}}_z = -\cos(\alpha) \cdot \cos(\beta)$

After the simulation is ran, the aerodynamic parameters are exported to the output CSV. Normal and Axial forces are converted to lift and drag and their respective coefficients through the following equations.

$$\begin{aligned}
\text{lift} &= \text{N} \cdot \cos(\alpha) - \text{A} \cdot \sin(\alpha) \cdot \cos(\beta) \\
\text{drag} &= \text{A} \cdot \cos(\alpha) \cdot \cos(\beta) + \text{N} \cdot \sin(\alpha) \\
\text{CL} &= \frac{\text{lift}}{0.5 \cdot \text{RHOREF} \cdot (\text{mach} \cdot \text{aRef})^2 \cdot \text{SREF}} \\
\text{CD} &= \frac{\text{drag}}{0.5 \cdot \text{RHOREF} \cdot (\text{mach} \cdot \text{aRef})^2 \cdot \text{SREF}}
\end{aligned}$$

The cases with mesh refinement use the following equation to determine which cells need to be refined. Cells above the range of [0.4, 0.6] were refined and cells below were coarsened. The boundaries of this range were chosen experimentally, chosen as values that would increase the cell count by a significant amount without having too large of an impact on solution time.

$$Refinement = |\nabla Mach\_Number| \cdot CellSize$$

2.4.2 Submit_star.pbs

This file is submitted to the PACE cluster with the command "qsub submit_star.pbs", instructing it to run the STAR-CCM+ simulations.

QSUB Parameters: Nodes, PPN, and Pmem control the amount of processors running the simulation, where total processes = Nodes * PPN. Pmem dictates the amount of memory that is given per process. Walltime is the total amount of time the simulation is allowed to be running, limited to 8hrs for the Embers backfill. In general, I targeted each simulation to run with 92 processes, upping the amount of processes if resources were available.

Environment Variables: MACROFILE tells the Pheonix Cluster what java file that will control the simulation. Likewise, SIMFILE is the path of the Star-CCM+ simulation file.

3 Results and Discussion

All four inviscid simulations were conducted in addition to a single viscous case, with results located in Appendix D. Additionally, full-sized copies of all of the graphs are located in Appendix E.

Mesh Convergence One critical concept in CFD is the idea of Mesh Convergence, or the idea that smaller and smaller meshes typically result in more precise solutions. Because mesh size directly trades with solution convergence speed, it is ideal to have as few cells as possible while still approaching an answer that is precise enough for the problem being solved. With finite computing resources in my situation, solution speed trades with the number of data points that can be acquired. It's worth considering if a slight decrease in solution precision would be worth it if we can capture more Mach numbers in transonic to more accurately capture the drag divergence relationship. These ideas will be investigated a little more thoroughly as I analyze the data.

Data Results With the first set of meshes, there were some issues with convergence in the transonic and supersonic cases. This is because I wasn't putting much detail in to the way I meshed in Pointwise with regards to refinement, resulting in cells with very high volume ratios and skewnesses. The second time around, I put far more effort into ensuring the surface mesh didn't have any cells with an area ratio above 20, which resulted in few cells of high volume ratio and skewness.

With the end results, there was only a single point in the fine mesh with mesh refinement case that converged to an erroneous value. This was replaced with extrapolation from a quadratic drag polar.

Drag Polars As expected for our inviscid simulations, we find a simple quadratic drag polars for all of our cases. Drag polars are a useful way to analyze mesh convergence because they display both lift and drag on a single chart. For subsonic cases, the fine meshes consistently converge to higher lift coefficients and lower drag coefficients, whereas the supersonic cases do the opposite. Mesh refinement had the largest effect for drag polars in the transonic range, reporting higher lift and lower drag. Otherwise, the effect of mesh refinement seems to be rather negligible.

Drag Divergence and Prandtl-Glauert As Anderson writes in Fundamentals of Aerodynamics [1], the aerodynamic properties of airfoils change drastically at Mach 1. More specifically, we expect the coefficients of drag and lift to increase almost asymptotically near Mach 1 and then return back to normal values. Indeed, Figure 6 shows how the lift and drag coefficients for the F-16 change around transonic region.

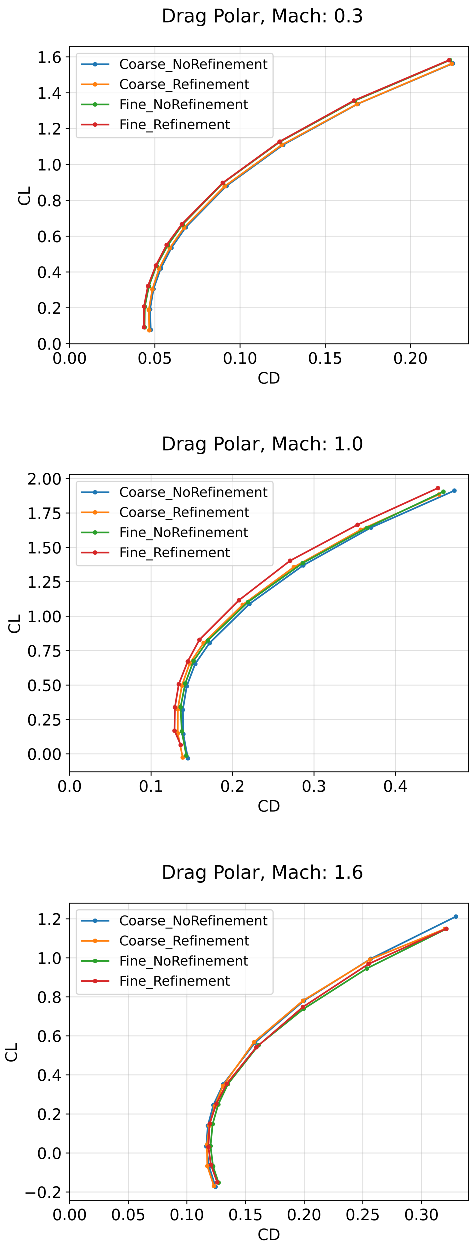


Figure 5: Figure

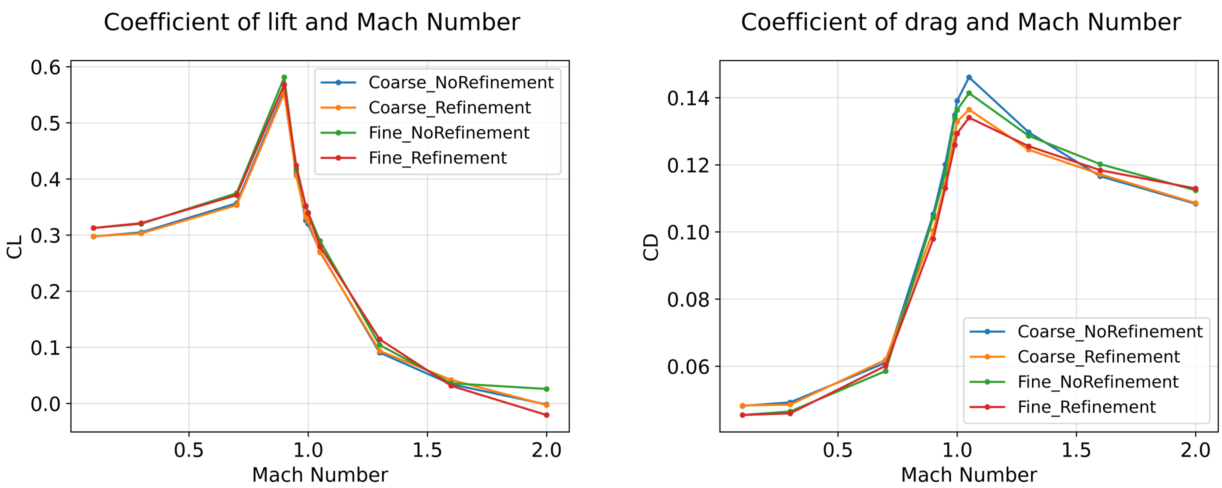


Figure 6: Lift and Drag relationship with Mach Number

Viscous Case One thing I was curious to look into is whether or not we would be able to see the phenomenon of a “drag bucket” with the F-16. I ran a case at Mach 0.3 with more angle of attack points to see if this phenomenon would show, using the K-epsilon model with no freestream turbulence. As seen in Figure 7, there does not seem to be a significant drag bucket. Why this may be the case I'd guess would be one (or the combination of) three possibilities. First, I could have set up the model wrong which is why there doesn't seem to be laminar flow. Second, that because the F-16's airfoil probably isn't optimized to have laminar flow, the aircraft doesn't have these properties. Or third, the skin friction from the large fuselage body surface area dominates the small decrease in laminar flow on the wing, making the existence of existence of a drag bucket negligible.

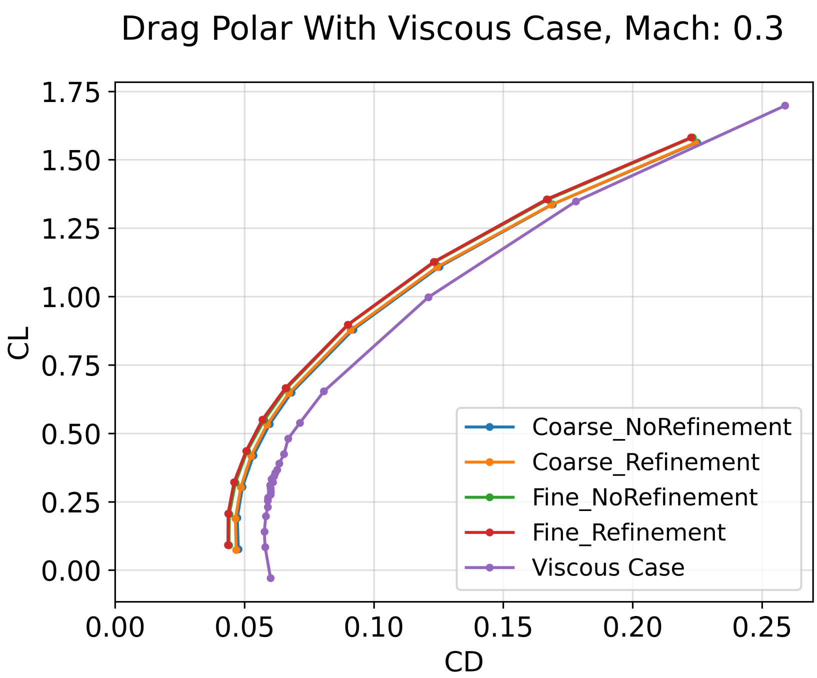


Figure 7: Drag Polar with Viscous Case

Other Results There are lots of other great data insights I found looking through the data. However, for brevity, I didn't include them in the main report. If interested, I encourage you to check them out in Appendix E.

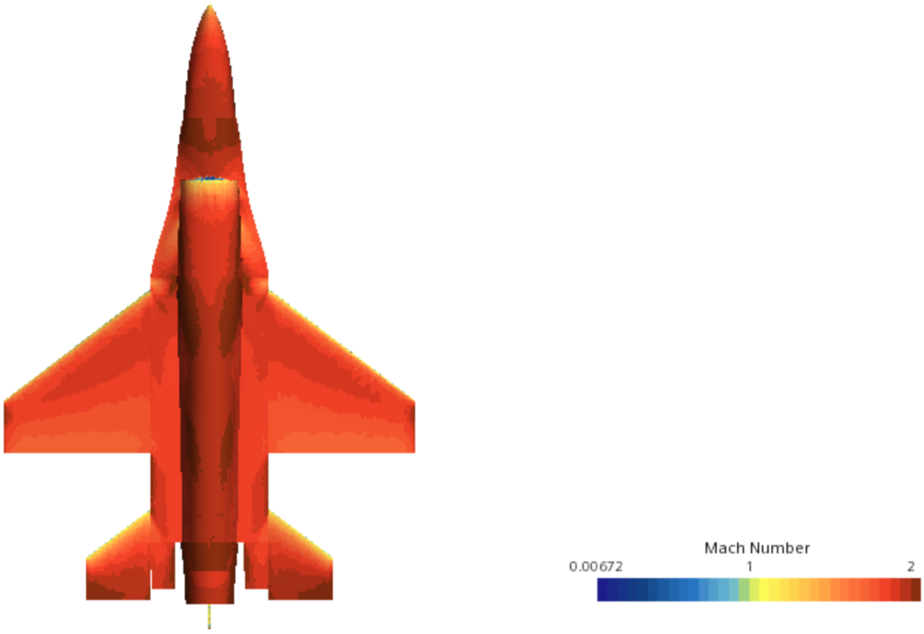


Figure 8: Mach Profile of Aircraft, $M_{inf} = 2, \alpha = 11$

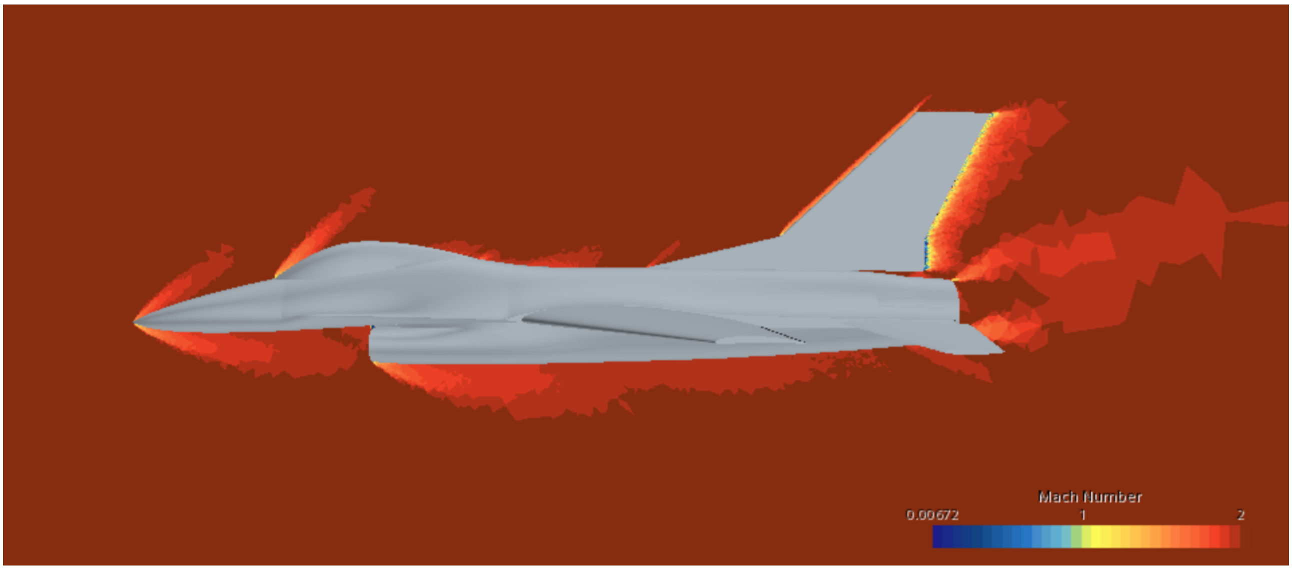


Figure 9: Mach Profile of Aircraft, $M_{inf} = 2, \alpha = 11$

4 Conclusion

The following report demonstrates a procedure that allows us to analyze the aerodynamic properties of a F-16 through a complete sweep of Mach numbers and angles of attack. The main conclusions are:

1. Mesh quality has significant effects on the lift and drag of an aircraft. While not perfect, a 960k cell mesh was able to decently capture the performance parameters of an F-16 in most Mach regimes, being weakest around transonic. In future work, it would be worth investigating an even more coarse mesh. Mesh refinement had a lower impact on my results than I expected, which given the amount of computing resources it used would have me consider it to be not worth it in future runs

2. The effects of Drag Divergence were quite significant and should be captured with even more test cases around Mach 1. Interestingly, these effects on lift occur a little earlier than Mach 1, with the largest differences occuring around Mach 0.8.

3. The laminar flow effects do not seem to be very present with the F-16 and would likely be better observed in a vehicle like a glider.

Overall, this project was a blast and I'm super happy to have proposed and followed through on it. The applications of CFD and related processes in industry are super cool, and as such, this project has definitely shaped my view on the type of work I want to go into.

References

[1] John Anderson. *Fundamentals of Aerodynamics*. McGraw-Hill, 2017.

[2] Cadence. Compute grid spacing for a given y+. URL: https://www.cadence.com/en_US/home/tools/system-analysis/computational-fluid-dynamics/y-plus.html.

[3] Sarthak Kinger. CFD analysis of F-16 Aircraft, June 28 2019. URL: https://grabcad.com/library/cfd-analysis-of-f-16-aircraft-1.

In no particular order, I would like to credit James Shields, Dean Ellis, Christian Perron, and Jai Ahuja for assisting me and providing me the skills to take on this project.

Appendix

A Generate Deck.java

```java
/*
 * Generates aerodynamic deck
 * Author: Carter Tegen
 * Written: December 2022
 */
package macro;
import java.util.*;
import java.io.*;
import star.common.*;
import star.base.neo.*;
import star.material.*;
import star.resurfacer.*;
import star.coupledflow.*;
import star.vis.*;
import star.flow.*;
import star.energy.*;
import star.metrics.*;
import star.meshing.*;
import star.mapping.*;

import java.time.format.DateTimeFormatter;
import java.time.LocalDateTime;

public class generateDeck extends StarMacro {

    static double[] machArray = {0.1, 0.3, 0.7, 0.9, 0.95, 0.99, 1, 1.05, 1.3, 1.6, 2};
    static double[] aoaArray = {-2, -1, 0, 1, 2, 3, 5, 7, 9, 11};
    static double[] betaArray = {0};


    int count = 1;
    int numIters = machArray.length * aoaArray.length * betaArray.length;

    final double TREF = 288.15;
    final double SREF = 39.58; //m2
    final double SVTREF = 5.92; //m2
    final double RHOREF = 1.225; //kg/m3
    final double aRef = 340;

    Simulation sim;

    String fileOut = "Fine_NoRefinement_Results";

    public void execute() {

        DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");
        LocalDateTime now = LocalDateTime.now();
        System.out.println(dtf.format(now));

        sim = getActiveSimulation();
        try {
```

```
        PrintWriter out = new PrintWriter(new FileWriter(new File(resolvePath(fileOut +
            ".csv"))));
        out.println("Mach, Beta, AOA, Lift, Drag, Moment, L/D, CL, CD");


        for(double mach : machArray) {
            for(double beta : betaArray) {
                updateParameters(calculateFlowAngle(aoaArray[0], betaArray[0]), mach);

                for(double aoa : aoaArray) {

                    sim.println(String.format("---On run %d of %d---", count, numIters));

                    double[] flowAngle = calculateFlowAngle(aoa, beta);
                    //sim.println("Updating parameters");
                    updateParameters(flowAngle, mach);
                    initialize();
                    //sim.println("Running case");
                    runCase();

                    //sim.println("Printing output");
                    printOutput(out, mach, aoa, beta);

                    count++;
                }
            }
        }

        out.close();
    } catch (Exception e) {

        sim.println("Error in execute");
    }
}

//Updates the farfield parameters to new Mach number and flow direction
public void updateParameters(double[] flowAngle, double mach) {
    PhysicsContinuum physicsContinuum_0 =
    ((PhysicsContinuum) sim.getContinuumManager().getContinuum("Physics 1"));

    VelocityProfile velocityProfile_0 =
    physicsContinuum_0.getInitialConditions().get(VelocityProfile.class);

    Units units_4 =
    ((Units) sim.getUnitsManager().getObject("m/s"));

    double xVelocity = mach*aRef*flowAngle[0];
    double yVelocity = mach*aRef*flowAngle[1];
    double zVelocity = mach*aRef*flowAngle[2];


    velocityProfile_0.getMethod(ConstantVectorProfileMethod.class).getQuantity().setComponentsAndUni
        yVelocity, zVelocity, units_4);

    Region region_0 =
    sim.getRegionManager().getRegion("fluid");
```

```
        Boundary boundary_1 =
        region_0.getBoundaryManager().getBoundary("farfield");

        FlowDirectionProfile flowDirectionProfile_0 =
        boundary_1.getValues().get(FlowDirectionProfile.class);

        Units units_3 =
        ((Units) sim.getUnitsManager().getObject(""));

        flowDirectionProfile_0.getMethod(ConstantVectorProfileMethod.class).getQuantity().setComponentsA
            flowAngle[1], flowAngle[2], units_3);

        MachNumberProfile machNumberProfile_0 =
        boundary_1.getValues().get(MachNumberProfile.class);

        machNumberProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValueAndUnits(
            units_3);
    }

    //Runs the simulation
    public void runCase() {
        Simulation sim =
        getActiveSimulation();

        Solution solution_0 =
        sim.getSolution();

        sim.getSimulationIterator().run();
    }

    //Initializes the simulation
    public void initialize() {
        Simulation sim =
        getActiveSimulation();

        Solution solution_0 =
        sim.getSolution();

        solution_0.clearSolution();

        solution_0.initializeSolution();
    }

    /*
     * Outputs line of data to csv
     */
    public void printOutput(PrintWriter fr, double mach, double aoa, double beta) {
        Simulation sim =
        getActiveSimulation();

        ForceReport forceReport_2 =
        ((ForceReport) sim.getReportManager().getReport("Axial Force"));

        MomentReport momentReport_1 =
        ((MomentReport) sim.getReportManager().getReport("Rotation Moment"));
```

```
    ForceReport forceReport_1 =
    ((ForceReport) sim.getReportManager().getReport("Normal Force"));


    sim.println("Getting monitored values");
    double normalForce = forceReport_1.monitoredValue();
    double axialForce = forceReport_2.monitoredValue();
    double moment = momentReport_1.monitoredValue();

    sim.println("Got monitored values");

    double caoa = Math.cos(Math.toRadians(aoa));
    double saoa = Math.sin(Math.toRadians(aoa));

    double cbeta = Math.cos(Math.toRadians(beta));
    double sbeta = Math.sin(Math.toRadians(beta));

    double lift = normalForce * caoa - axialForce * saoa * cbeta;
    double drag = axialForce * caoa * cbeta + normalForce * saoa;

    double CL = lift/(0.5 * RHOREF * Math.pow((mach * aRef), 2) * SREF);
    double CD = drag/(0.5 * RHOREF * Math.pow((mach * aRef), 2) * SREF);

    sim.println("Calculated lift and drag");

    //Mach, Beta, AOA, Lift, Drag, Moment, L/D
    String outString = String.format("%f, %f, %f, %f, %f, %f, %f, %f, %f",
       mach, beta, aoa, lift, drag, moment, lift/drag, CL, CD);

    sim.println("Outstring ready");

    try {
       fr.println(outString);
    } catch (Exception e) {
       sim.print("Error in printOutput");
    }

    sim.println("Printed");


}

/*
 * For an inputted angle of attack and sideslip, calculates the necessary flow
 * direction
 */
public double[] calculateFlowAngle(double aoa, double beta) {
    double x = Math.sin(Math.toRadians(beta));
    double y = Math.sin(Math.toRadians(aoa)) * Math.cos(Math.toRadians(beta));
    double z = -Math.cos(Math.toRadians(aoa)) * Math.cos(Math.toRadians(beta));

    double[] toRet = {x,y,z};

    return toRet;
}
```

```
}
```

B Submit_star.pbs

```
#!/bin/bash
# ----------------QSUB Parameters----------------- #
#PBS -N CCM_Test
#PBS -A GT-jt59-FAA_A10_CostShare
#PBS -l nodes=3:ppn=24,pmem=6gb
#PBS -l walltime=6:00:00
#PBS -q embers
#PBS -j oe
#PBS -o pbs_job.out
#PBS -m abe
#PBS -M cartertegen@gatech.edu

# ----------------Load Modules-------------------- #
module load openmpi/3.1.6
module load starccmplus/17.02.007

export OPENMPI_DIR=$OPENMPI_ROOT

# -------------Environment Variables--------------- #
cd ${PBS_O_WORKDIR}

MACROFILE="generateDeck.java"
SIMFILE="f16sim_noRefinement_fine.sim"

# ----------------Print Some Info-------------------- #
echo Running on host `hostname`
echo Time is `date`
echo Directory is `pwd`
echo This jobs runs on the following processors:
NODES=`cat $PBS_NODEFILE`
echo $NODES
# Compute the number of processors
echo This job has allocated ${PBS_NP} nodes

# ---------------Execute Script ------------------ #
starccm+ -mpi openmpi -licpath 1999@ugslic2.ecs.gatech.edu -machinefile ${PBS_NODEFILE} -np
    ${PBS_NP} -batch $MACROFILE $SIMFILE > starccm_${PBS_JOBID}.log
```

C Analyze_Data.py

```
import matplotlib.pyplot as plt
import numpy as np
import csv


filePaths = ["Coarse_NoRefinement_Results.csv", \
            "Coarse_Refinement_Results.csv", \
            "Fine_NoRefinement_Results.csv", \
               "Fine_Refinement_Results.csv"]

numFiles = len(filePaths)
rows = 110
cols = 9

viscid_rows = 25
viscid_cols = 9

d = {"Mach": 0,\
     "Beta": 1,\
     "Alpha": 2,\
     "Lift": 3,\
     "Drag": 4,\
     "Moment": 5,\
     "LD": 6,\
     "CL": 7,\
     "CD": 8}

data = np.empty([numFiles, rows, cols])
viscous_data = np.empty([1, viscid_rows, viscid_cols])

def importData():
    for i, fileP in enumerate(filePaths):
        data[i,:,:] = np.loadtxt(fileP, delimiter=",", skiprows=1)
        #print(data[i])

def importViscousData():
    viscous_data[0,:,:] = np.loadtxt("Viscous_Results.csv", delimiter=",", skiprows=1)

def createPlots():
    plt.rcParams.update({'font.size': 14})
    plt.rc('legend', fontsize=12)

    plt.figure(1)
    dragPolar(Mach = 0.3)
    plt.savefig("Photos/DragPolarM0_3.png", dpi=600)

    plt.figure(2)
    dragPolar(Mach = 1)
    plt.savefig("Photos/DragPolarM1.png", dpi=600)

    plt.figure(3)
    dragPolar(Mach = 1.6)
```

```
    plt.savefig("Photos/DragPolarM1_6.png", dpi=600)

    plt.figure(4)
    liftMach()
    plt.savefig("Photos/LiftAndMach.png", dpi=600)

    plt.figure(5)
    dragMach()
    plt.savefig("Photos/DragAndMach.png", dpi=600)

    plt.figure(6)
    liftCurve(Mach = 0.3)
    plt.savefig("Photos/LiftCurveM0_3.png", dpi=600)

    plt.figure(7)
    liftCurve(Mach = 1)
    plt.savefig("Photos/LiftCurveM1.png", dpi=600)

    plt.figure(8)
    liftCurve(Mach = 1.6)
    plt.savefig("Photos/LiftCurveM1_6.png", dpi=600)

    plt.figure(9)
    LDCurveAOA(Mach = 0.3)
    plt.savefig("Photos/LDCurveM0_3.png", dpi=600)

    plt.figure(10)
    LDCurveAOA(Mach = 1)
    plt.savefig("Photos/LDCurveM1.png", dpi=600)

    plt.figure(11)
    LDCurveAOA(Mach = 1.6)
    plt.savefig("Photos/LDCurveM1_6.png", dpi=600)

    plt.figure(12)
    LDCurveMach(0)
    plt.savefig("Photos/LDCurveMach_0deg", dpi=600)

    plt.figure(13)
    LDCurveMach(3)
    plt.savefig("Photos/LDCurveMach_3deg", dpi=600)

    plt.figure(14)
    LDCurveMach(11)
    plt.savefig("Photos/LDCurveMach_11deg", dpi=600)

    plt.figure(15)
    dragPolarAndViscous()
    plt.savefig("Photos/DragPolarAndViscousM0_3.png", dpi=600)

    plt.show()


def dragPolar(Mach = 0.1):
    for i in range(numFiles):
        plt.plot(data[i, data[i, :, d["Mach"]] == Mach, d["CD"]], \
```

```
        data[i, data[i, :, d["Mach"]] == Mach, d["CL"]],
        label=filePaths[i][:-12], marker = '.')

    plt.xlabel("CD")
    plt.ylabel("CL")
    plt.xlim(left=0)
    plt.legend()
    plt.grid(alpha = 0.4)
    plt.suptitle("Drag Polar, Mach: %.1f" % Mach )

def dragPolarAndViscous(Mach = 0.3):
    for i in range(numFiles):
        plt.plot(data[i, data[i, :, d["Mach"]] == Mach, d["CD"]], \
            data[i, data[i, :, d["Mach"]] == Mach, d["CL"]],
            label=filePaths[i][:-12], marker = '.')

    print(viscous_data)
    plt.plot(viscous_data[0, 1:, d["CD"]], viscous_data[0, 1:, d["CL"]],
            label="Viscous Case", marker = '.')

    plt.xlabel("CD")
    plt.ylabel("CL")
    plt.xlim(left=0)
    plt.legend()
    plt.grid(alpha = 0.4)
    plt.suptitle("Drag Polar With Viscous Case, Mach: %.1f" % Mach )

def liftMach(aoa = 0):
    for i in range(numFiles):
        mask = data[i,:,d["Alpha"]] == aoa
        plt.plot(data[i, mask, d["Mach"]], data[i, mask, d["CL"]], \
            label=filePaths[i][:-12], marker = '.')

    plt.legend()
    plt.xlabel("Mach Number")
    plt.ylabel("CL")
    plt.grid(alpha = 0.4)
    plt.suptitle("Coefficient of lift and Mach Number")

def dragMach(aoa = 0):
    for i in range(numFiles):
        mask = data[i,:,d["Alpha"]] == aoa
        plt.plot(data[i, mask, d["Mach"]], data[i, mask, d["CD"]], \
             label=filePaths[i][:-12], marker = '.')

    plt.legend()
    plt.xlabel("Mach Number")
    plt.ylabel("CD")
    plt.grid(alpha = 0.4)
    plt.suptitle("Coefficient of drag and Mach Number")

def liftCurve(Mach = 0.3):
    for i in range(numFiles):
        mask = data[i, :, d["Mach"]] == Mach
        mask
        plt.plot(data[i, mask, d["Alpha"]], data[i, mask, d["CL"]], \
```

```
            label = filePaths[i][:-12], marker = '.')

    plt.legend()
    plt.xlabel("Angle of Attack")
    plt.ylabel("CL")
    plt.grid(alpha = 0.4)
    plt.suptitle("Lift Curve, Mach: %.1f" % Mach )

def LDCurveAOA(Mach = 0.3):
    for i in range(numFiles):
        mask = data[i, :, d["Mach"]] == Mach
        plt.plot(data[i, mask, d["Alpha"]], data[i, mask, d["LD"]], \
            label = filePaths[i][:-12], marker = '.')

    plt.legend()
    plt.xlabel("Angle of Attack")
    plt.ylabel("L/D")
    plt.grid(alpha = 0.4)
    plt.suptitle("Lift/Drag Curve wrt Alpha, Mach: %.1f" % Mach )

def LDCurveMach(aoa = 3):
    for i in range(numFiles):
        mask = data[i, :, d["Alpha"]] == aoa
        mask
        plt.plot(data[i, mask, d["Mach"]], data[i, mask, d["LD"]], \
            label = filePaths[i][:-12], marker = '.')

    plt.legend()
    plt.xlabel("Mach Number")
    plt.ylabel("L/D")
    plt.grid(alpha = 0.4)
    plt.suptitle("Lift/Drag Curve wrt Mach, Angle of Attack: %.1f" % aoa )


if __name__ == "__main__":
    importData()
    importViscousData()
    createPlots()
```

D Results Tables

D.1 Coarse Mesh, No Refinement

Mach	Beta	AOA	Lift (N)	Drag (N)	L/D	CL	CD
0.1	0	-2	2043.072108	1311.466122	1.557854	0.072903	0.046797
0.1	0	-1	5208.96032	1300.486764	4.005393	0.185871	0.046405
0.1	0	0	8335.295656	1350.226736	6.173256	0.297428	0.04818
0.1	0	1	11492.8079	1466.168924	7.838666	0.410097	0.052317
0.1	0	2	14649.23734	1634.083959	8.964801	0.522727	0.058309
0.1	0	3	17817.78245	1861.020607	9.574199	0.63579	0.066407
0.1	0	5	24185.73295	2505.959839	9.651285	0.863017	0.08942
0.1	0	7	30517.97664	3416.142775	8.933461	1.08897	0.121898
0.1	0	9	36801.25026	4619.012763	7.967341	1.313176	0.16482
0.1	0	11	43064.86458	6137.718811	7.016428	1.53668	0.219012
0.3	0	-2	19367.82506	12011.8235	1.612397	0.076789	0.047624
0.3	0	-1	48207.52534	11909.68873	4.047757	0.191132	0.047219
0.3	0	0	76879.18418	12420.78366	6.18956	0.304808	0.049246
0.3	0	1	105751.9071	13492.58266	7.837781	0.419282	0.053495
0.3	0	2	134664.9733	15068.82805	8.936659	0.533915	0.059744
0.3	0	3	163662.9885	17186.47585	9.522778	0.648886	0.06814
0.3	0	5	221857.5598	23210.08859	9.558669	0.879614	0.092023
0.3	0	7	279751.9919	31612.55405	8.849395	1.109152	0.125336
0.3	0	9	337262.2897	42637.9155	7.909915	1.337167	0.169049
0.3	0	11	394241.4655	56716.93168	6.951037	1.563076	0.224869
0.7	0	-2	143356.6584	79158.53793	1.811007	0.104396	0.057645
0.7	0	-1	316679.6979	79603.28973	3.978224	0.230613	0.057969
0.7	0	0	489750.3088	83846.79853	5.841014	0.356647	0.061059
0.7	0	1	663728.9107	91514.79416	7.252695	0.483342	0.066643
0.7	0	2	838367.3016	102683.3527	8.164588	0.610518	0.074776
0.7	0	3	1013880.023	117309.3905	8.642787	0.73833	0.085427
0.7	0	5	1367814.915	157556.3268	8.681434	0.996074	0.114736
0.7	0	7	1722548.85	213696.1211	8.06074	1.254399	0.155618
0.7	0	9	2076359.255	290496.5725	7.14762	1.512052	0.211546
0.7	0	11	2410385.94	390656.1995	6.170095	1.755298	0.284485
0.9	0	-2	542578.7477	215095.7838	2.522498	0.239022	0.094756
0.9	0	-1	909915.17	222065.6094	4.097506	0.400845	0.097827
0.9	0	0	1265258.775	238970.6423	5.29462	0.557384	0.105274
0.9	0	1	1626894.567	273431.8886	5.949908	0.716695	0.120455
0.9	0	2	1946910.792	312782.8749	6.22448	0.857672	0.13779
0.9	0	3	2249611.248	363114.2396	6.195326	0.991021	0.159963
0.9	0	5	2857505.081	481165.675	5.938713	1.258816	0.211968
0.9	0	7	3460684.046	643923.2298	5.374374	1.524534	0.283667
0.9	0	9	4050570.977	837706.4648	4.835311	1.784397	0.369035
0.9	0	11	4589445.01	1060630.259	4.327092	2.021787	0.467239
0.95	0	-2	247627.6522	310645.9476	0.797138	0.097907	0.122823
0.95	0	-1	647638.3375	300908.847	2.152274	0.256062	0.118973
0.95	0	0	1040311.366	303716.474	3.425271	0.411317	0.120083

0.95	0	1	1405292.142	322881.593	4.352345	0.555622	0.12766
0.95	0	2	1768418.497	357429.0155	4.947608	0.699195	0.14132
0.95	0	3	2118221.871	404287.9706	5.239389	0.837499	0.159847
0.95	0	5	2812182.848	527331.4247	5.332857	1.111877	0.208496
0.95	0	7	3499355.605	689078.7542	5.07831	1.38357	0.272447
0.95	0	9	4168756.552	894617.1196	4.659822	1.648237	0.353712
0.95	0	11	4840573.28	1149587.371	4.210705	1.913859	0.454522
0.99	0	-2	-76730.56079	383306.9263	-0.20018	-0.027936	0.139552
0.99	0	-1	413923.3107	371897.7609	1.113003	0.150699	0.135398
0.99	0	0	895536.5508	370322.9554	2.418258	0.326042	0.134825
0.99	0	1	1365623.164	383114.8267	3.564527	0.497188	0.139482
0.99	0	2	1802109.747	413066.3043	4.362761	0.656102	0.150387
0.99	0	3	2206497.532	460461.1325	4.79193	0.803329	0.167642
0.99	0	5	2990174.304	594965.1624	5.025797	1.088645	0.216611
0.99	0	7	3754137.503	775139.1768	4.843179	1.366785	0.282208
0.99	0	9	4520819.755	1006745.152	4.49053	1.645914	0.36653
0.99	0	11	5273331.803	1294610.447	4.073296	1.919884	0.471334
1	0	-2	-89962.97479	406981.7453	-0.221049	-0.032101	0.145223
1	0	-1	401025.1294	391343.9692	1.024738	0.143097	0.139643
1	0	0	896373.3078	389717.3707	2.30006	0.319852	0.139063
1	0	1	1379188.711	403147.4542	3.421053	0.492135	0.143855
1	0	2	1832255.499	432172.8886	4.239635	0.653802	0.154212
1	0	3	2255613.531	481480.7676	4.684743	0.804869	0.171806
1	0	5	3051349.155	618609.6853	4.932592	1.08881	0.220738
1	0	7	3837126.539	803710.2436	4.774266	1.369198	0.286787
1	0	9	4606524.299	1037144.251	4.441546	1.643742	0.370083
1	0	11	5358275.837	1323644.301	4.048124	1.911989	0.472315
1.05	0	-2	-183962.5281	467420.1122	-0.39357	-0.05954	0.151283
1.05	0	-1	320949.9255	452383.447	0.709464	0.103877	0.146416
1.05	0	0	836722.0837	451273.1423	1.854137	0.270809	0.146057
1.05	0	1	1348305.831	464127.989	2.90503	0.436385	0.150217
1.05	0	2	1863258.162	493339.4972	3.776827	0.603052	0.159672
1.05	0	3	2354099.4	537327.4188	4.381127	0.761915	0.173908
1.05	0	5	3239649.147	679633.8674	4.766756	1.048527	0.219967
1.05	0	7	4105148.566	867111.3824	4.734281	1.32865	0.280645
1.05	0	9	4920989.603	1113913.111	4.41775	1.592701	0.360523
1.05	0	11	5726354.525	1419473.352	4.03414	1.853361	0.459419
1.3	0	-2	-840987.6468	650816.0544	-1.292205	-0.177567	0.137414
1.3	0	-1	-208903.414	622127.9723	-0.335788	-0.044108	0.131357
1.3	0	0	428371.1103	614226.3227	0.697416	0.090447	0.129689
1.3	0	1	1071717.112	626519.9949	1.710587	0.226284	0.132284
1.3	0	2	1724323.312	659883.0668	2.613074	0.364076	0.139329
1.3	0	3	2380639.444	714575.9461	3.331542	0.502652	0.150877
1.3	0	5	3694713.66	888292.5187	4.159343	0.780107	0.187555
1.3	0	7	4990624.116	1146792.429	4.351811	1.053728	0.242135
1.3	0	9	6273838.152	1493298.344	4.201329	1.324668	0.315297
1.3	0	11	7503710.873	1919920.207	3.908345	1.584345	0.405375

1.6	0	-2	-1245392.763	892706.7319	-1.395075	-0.173591	0.124431
1.6	0	-1	-493553.4637	853117.7023	-0.578529	-0.068795	0.118913
1.6	0	0	244923.6061	836434.3926	0.292819	0.034139	0.116588
1.6	0	1	997256.2079	845777.5322	1.1791	0.139004	0.11789
1.6	0	2	1756767.268	879592.9833	1.99725	0.244869	0.122603
1.6	0	3	2519497.118	939516.5699	2.681695	0.351184	0.130956
1.6	0	5	4057401.134	1134389.543	3.576726	0.565546	0.158118
1.6	0	7	5600408.922	1434642.581	3.903696	0.780621	0.19997
1.6	0	9	7139440.884	1842791.235	3.874254	0.995141	0.25686
1.6	0	11	8689308.118	2363036.443	3.677179	1.211171	0.329375
2	0	-2	-1889960.759	1308420.706	-1.44446	-0.168598	0.116721
2	0	-1	-966540.8655	1251589.833	-0.77225	-0.086222	0.111651
2	0	0	-22221.23012	1215025.864	-0.018289	-0.001982	0.108389
2	0	1	958767.1731	1236375.966	0.775466	0.085529	0.110294
2	0	2	1760846.254	1274459.325	1.381642	0.15708	0.113691
2	0	3	3023253.446	1365680.082	2.213735	0.269696	0.121829
2	0	5	4700612.108	1558661.506	3.0158	0.419329	0.139044
2	0	7	6591122.737	1917954.961	3.436537	0.587976	0.171096
2	0	9	8310291.209	2375526.969	3.498294	0.741338	0.211914
2	0	11	10317073.03	3027586.317	3.407689	0.920358	0.270083

D.2 Coarse Mesh, Refinement

Mach	Beta	AOA	Lift	Drag	L/D	CL	CD
0.1	0	-2	2044.243582	1309.712931	1.560833	0.072945	0.046734
0.1	0	-1	5199.820605	1296.824566	4.009656	0.185545	0.046274
0.1	0	0	8348.872857	1354.039157	6.165902	0.297912	0.048316
0.1	0	1	11508.96485	1465.91659	7.851037	0.410673	0.052308
0.1	0	2	14674.0775	1634.888784	8.975581	0.523614	0.058338
0.1	0	3	17846.13651	1862.209142	9.583315	0.636802	0.066449
0.1	0	5	24216.24128	2505.469281	9.665352	0.864106	0.089402
0.1	0	7	30553.00029	3422.662058	8.926677	1.09022	0.122131
0.1	0	9	36857.91173	4623.176538	7.972421	1.315198	0.164968
0.1	0	11	43130.24445	6149.566993	7.013542	1.539013	0.219434
0.3	0	-2	18773.85502	11801.43499	1.590811	0.074434	0.04679
0.3	0	-1	47577.35391	11726.99463	4.05708	0.188633	0.046495
0.3	0	0	76395.23876	12248.71209	6.237002	0.302889	0.048563
0.3	0	1	105400.4784	13249.61502	7.954984	0.417888	0.052532
0.3	0	2	134216.5225	14810.62487	9.062178	0.532137	0.058721
0.3	0	3	163168.9591	16941.87914	9.631102	0.646927	0.067171
0.3	0	5	221359.1472	22943.64794	9.647949	0.877638	0.090966
0.3	0	7	279456.5532	31377.62483	8.906237	1.10798	0.124405
0.3	0	9	336929.4486	42511.14416	7.925673	1.335847	0.168547
0.3	0	11	393931.1436	56534.14556	6.968022	1.561846	0.224145
0.7	0	-2	139990.2868	78278.73733	1.788356	0.101944	0.057004
0.7	0	-1	313510.714	78496.36159	3.993952	0.228306	0.057163
0.7	0	0	484982.4854	85028.05801	5.703794	0.353175	0.061919
0.7	0	1	659850.1819	90496.16631	7.291471	0.480518	0.065901
0.7	0	2	833237.3946	96741.19602	8.613057	0.606782	0.070449
0.7	0	3	1008225.391	118017.9106	8.542986	0.734213	0.085943
0.7	0	5	1357681.122	161326.0424	8.415759	0.988694	0.117481
0.7	0	7	1701304.243	217370.9655	7.826732	1.238928	0.158294
0.7	0	9	2069191.145	292272.5657	7.079663	1.506832	0.21284
0.7	0	11	2416641.593	384693.7563	6.281988	1.759853	0.280143
0.9	0	-2	544988.5602	206628.2748	2.637531	0.240084	0.091026
0.9	0	-1	898783.169	210135.154	4.277167	0.395941	0.092571
0.9	0	0	1255373.675	227826.832	5.51021	0.553029	0.100365
0.9	0	1	1586104.845	257845.3263	6.151381	0.698726	0.113589
0.9	0	2	1914155.559	301525.0319	6.348248	0.843243	0.132831
0.9	0	3	2227104.918	350144.4437	6.360532	0.981106	0.154249
0.9	0	5	2839873.986	471907.7249	6.017859	1.251049	0.207889
0.9	0	7	3446590.87	628926.3586	5.480118	1.518326	0.277061
0.9	0	9	3979249.634	804526.6072	4.946076	1.752978	0.354418
0.9	0	11	4537239.188	1032695.167	4.39359	1.998789	0.454933
0.95	0	-2	229635.7366	295736.9804	0.776486	0.090793	0.116928
0.95	0	-1	639418.335	285448.73	2.240046	0.252812	0.11286
0.95	0	0	1027231.956	288966.1094	3.554853	0.406145	0.114251
0.95	0	1	1403599.445	306801.3351	4.574946	0.554953	0.121303

0.95	0	2	1768339.401	342826.8213	5.158113	0.699163	0.135546
0.95	0	3	2113911.356	390124.6938	5.418553	0.835795	0.154247
0.95	0	5	2800695.1	514971.568	5.438543	1.107335	0.203609
0.95	0	7	3480207.589	678340.7721	5.130471	1.375999	0.268201
0.95	0	9	4158186.861	886069.4319	4.692845	1.644058	0.350333
0.95	0	11	4809220.736	1137678.533	4.227223	1.901463	0.449814
0.99	0	-2	-63985.91003	369982.5596	-0.172943	-0.023296	0.134701
0.99	0	-1	431695.8586	355948.5009	1.212804	0.157169	0.129592
0.99	0	0	911850.4522	354824.5091	2.569863	0.331981	0.129182
0.99	0	1	1380464.335	369636.6243	3.734652	0.502591	0.134575
0.99	0	2	1818090.727	398460.6189	4.562786	0.66192	0.145069
0.99	0	3	2214559.486	446387.3518	4.961071	0.806264	0.162518
0.99	0	5	2980179.498	577741.447	5.158327	1.085006	0.210341
0.99	0	7	3728315.597	753608.2791	4.947286	1.357383	0.274369
0.99	0	9	4480930.156	970849.6987	4.615473	1.631391	0.353461
0.99	0	11	5175734.278	1248184.326	4.146611	1.884351	0.454432
1	0	-2	-71316.62091	389102.2289	-0.183285	-0.025448	0.138843
1	0	-1	419129.1404	372144.6638	1.126253	0.149557	0.132792
1	0	0	910265.4495	372365.6632	2.444547	0.324809	0.132871
1	0	1	1389641.606	386385.1073	3.59652	0.495865	0.137873
1	0	2	1838836.629	413101.2653	4.451297	0.65615	0.147407
1	0	3	2256737.952	460906.1672	4.896307	0.80527	0.164465
1	0	5	3035629.343	595702.0243	5.095886	1.083201	0.212564
1	0	7	3800815.597	771298.5154	4.927814	1.356242	0.275222
1	0	9	4561648.85	1001616.323	4.554288	1.627729	0.357406
1	0	11	5279639.8	1270746.367	4.154755	1.883929	0.453439
1.05	0	-2	-156630.4499	444305.0825	-0.352529	-0.050694	0.143801
1.05	0	-1	332903.4139	421987.5788	0.788894	0.107746	0.136578
1.05	0	0	831393.4891	421534.4393	1.972303	0.269084	0.136432
1.05	0	1	1354505.78	436685.2052	3.10179	0.438392	0.141335
1.05	0	2	1870071.526	466551.689	4.008284	0.605257	0.151002
1.05	0	3	2371862.695	511263.5917	4.639217	0.767664	0.165473
1.05	0	5	3261899.647	649946.6331	5.018719	1.055729	0.210358
1.05	0	7	4123583.498	844628.5276	4.882127	1.334616	0.273368
1.05	0	9	5001725.024	1093418.114	4.574394	1.618831	0.35389
1.05	0	11	5749948.743	1390207.085	4.136038	1.860997	0.449947
1.3	0	-2	-839110.5792	631693.1549	-1.328352	-0.177171	0.133377
1.3	0	-1	-203176.6587	603463.4888	-0.336684	-0.042899	0.127416
1.3	0	0	442838.1889	589646.0316	0.751024	0.093502	0.124499
1.3	0	1	1076316.837	606197.5538	1.775522	0.227255	0.127993
1.3	0	2	1731693.057	641786.6996	2.698238	0.365632	0.135508
1.3	0	3	2398693.061	691853.1408	3.467055	0.506464	0.146079
1.3	0	5	3701539.124	865381.6225	4.277349	0.781548	0.182718
1.3	0	7	4993470.942	1124089.641	4.442236	1.054329	0.237342
1.3	0	9	6255391.987	1466504.246	4.265512	1.320773	0.30964
1.3	0	11	7429356.734	1887740.975	3.935581	1.568645	0.39858
1.6	0	-2	-1214926.879	883201.9858	-1.375593	-0.169344	0.123106

1.6	0	-1	-480698.6252	842742.9296	-0.570398	-0.067003	0.117467
1.6	0	0	300074.6068	840676.0761	0.356944	0.041826	0.117179
1.6	0	1	1003022.033	855086.2763	1.173007	0.139808	0.119187
1.6	0	2	1724435.439	888077.8516	1.941762	0.240363	0.123786
1.6	0	3	2467240.386	938432.9223	2.629107	0.3439	0.130805
1.6	0	5	4076031.461	1128638.125	3.61146	0.568143	0.157317
1.6	0	7	5598211.525	1427864.474	3.920688	0.780314	0.199025
1.6	0	9	7109790.977	1837790.727	3.868662	0.991008	0.256163
1.6	0	11	8235914.625	2295867.956	3.587277	1.147974	0.320013
2	0	-2	-1650325.967	1321382.078	-1.248939	-0.147221	0.117877
2	0	-1	-959458.6218	1246188.464	-0.769915	-0.085591	0.111169
2	0	0	-33400.90032	1217487.597	-0.027434	-0.00298	0.108609
2	0	1	870980.6252	1249204.755	0.697228	0.077698	0.111438
2	0	2	1882807.337	1283534.118	1.466893	0.16796	0.114501
2	0	3	2927618.379	1349043.05	2.170145	0.261165	0.120344
2	0	5	4833053.614	1593734.875	3.032533	0.431144	0.142173
2	0	7	6469089.509	1922380.66	3.365145	0.57709	0.17149
2	0	9	8340523.783	2392204.41	3.486543	0.744035	0.213402
2	0	11	10533279.89	3142437.731	3.351945	0.939645	0.280328

D.3 Fine Mesh, No Refinement

Mach	Beta	AOA	Lift	Drag	L/D	CL	CD
0.1	0	-2	2444.070633	1212.360555	2.01596	0.087212	0.043261
0.1	0	-1	5599.492044	1214.930367	4.6089	0.199806	0.043352
0.1	0	0	8758.546238	1274.686637	6.871137	0.31253	0.045485
0.1	0	1	11921.42258	1396.53645	8.536421	0.425391	0.049832
0.1	0	2	15097.37996	1570.754985	9.611544	0.538718	0.056049
0.1	0	3	18283.02507	1808.172073	10.11133	0.652392	0.064521
0.1	0	5	24668.3959	2457.96719	10.036096	0.88024	0.087707
0.1	0	7	31013.95092	3367.18324	9.210651	1.106668	0.120151
0.1	0	9	37355.29523	4565.377729	8.182301	1.332946	0.162906
0.1	0	11	43599.70863	6098.361258	7.149414	1.555765	0.217607
0.3	0	-2	23011.22552	11096.98525	2.073647	0.091234	0.043997
0.3	0	-1	51832.02303	11135.1623	4.654806	0.205502	0.044148
0.3	0	0	80735.60528	11735.00246	6.879897	0.320098	0.046527
0.3	0	1	109570.157	12862.00154	8.518904	0.43442	0.050995
0.3	0	2	138597.4494	14544.60066	9.529134	0.549507	0.057666
0.3	0	3	167731.2461	16728.1818	10.026867	0.665015	0.066323
0.3	0	5	226131.2092	22714.24215	9.955481	0.896558	0.090057
0.3	0	7	284211.7214	31165.18612	9.119526	1.126834	0.123563
0.3	0	9	341774.871	42206.82628	8.097621	1.355058	0.16734
0.3	0	11	398972.4776	56294.52725	7.087234	1.581833	0.223195
0.7	0	-2	167405.1438	73830.02535	2.26744	0.121908	0.053765
0.7	0	-1	341106.4264	74662.57934	4.56864	0.248401	0.054371
0.7	0	0	514434.4823	80425.58566	6.396403	0.374623	0.058568
0.7	0	1	688097.0589	91049.08921	7.557429	0.501088	0.066304
0.7	0	2	864283.2864	100842.7724	8.570602	0.629391	0.073436
0.7	0	3	1031405.519	125852.8131	8.195331	0.751093	0.091649
0.7	0	5	1396473.895	157990.2122	8.83899	1.016944	0.115052
0.7	0	7	1751723.091	217696.7594	8.04662	1.275644	0.158532
0.7	0	9	2109872.988	292405.0892	7.215582	1.536457	0.212936
0.7	0	11	2430031.522	397920.4739	6.106827	1.769604	0.289775
0.9	0	-2	613162.1846	213839.5853	2.867393	0.270116	0.094203
0.9	0	-1	969476.4147	217563.028	4.456072	0.427083	0.095843
0.9	0	0	1319342.891	236826.4327	5.570928	0.58121	0.104329
0.9	0	1	1646605.555	270463.3555	6.088091	0.725379	0.119147
0.9	0	2	1969752.995	313127.6012	6.290576	0.867735	0.137942
0.9	0	3	2277928.012	361578.4402	6.299955	1.003495	0.159286
0.9	0	5	2889065.785	483499.1449	5.975328	1.272719	0.212996
0.9	0	7	3493643.947	639474.1967	5.463307	1.539054	0.281707
0.9	0	9	4056380.557	825133.5583	4.916029	1.786956	0.363496
0.9	0	11	4517235.502	1025668.017	4.404189	1.989977	0.451837
0.95	0	-2	238758.5701	304481.2789	0.784149	0.0944	0.120385
0.95	0	-1	656070.7501	293637.4074	2.234289	0.259396	0.116098
0.95	0	0	1056613.897	297056.1541	3.55695	0.417762	0.11745
0.95	0	1	1440040.787	314241.1119	4.582598	0.569361	0.124244

0.95	0	2	1814760.946	351590.4295	5.161577	0.717517	0.139011
0.95	0	3	2153228.533	398220.9614	5.40712	0.85134	0.157448
0.95	0	5	2838082.341	524210.8229	5.414009	1.122117	0.207262
0.95	0	7	3553247.886	685400.8704	5.184189	1.404878	0.270993
0.95	0	9	4197064.032	898488.9889	4.671247	1.659429	0.355243
0.95	0	11	4842240.438	1152714.466	4.200728	1.914518	0.455759
0.99	0	-2	-17284.69597	385378.0802	-0.044851	-0.006293	0.140306
0.99	0	-1	476629.7858	369747.1875	1.289069	0.173529	0.134615
0.99	0	0	964203.9517	368251.2541	2.618332	0.351042	0.134071
0.99	0	1	1426572.106	381555.2978	3.738834	0.519378	0.138914
0.99	0	2	1868798.291	411190.6862	4.544846	0.680381	0.149704
0.99	0	3	2270432.562	459524.7548	4.940828	0.826606	0.167301
0.99	0	5	3061538.732	593250.7372	5.160615	1.114627	0.215987
0.99	0	7	3822362.917	783132.0504	4.880866	1.391624	0.285118
0.99	0	9	4501764.336	990810.5496	4.543517	1.638976	0.360729
0.99	0	11	5215445.9	1265059.203	4.122689	1.898809	0.460575
1	0	-2	-34898.22374	401679.66	-0.086881	-0.012453	0.143331
1	0	-1	459997.9827	386104.9216	1.191381	0.164141	0.137773
1	0	0	952425.2102	381953.4474	2.493564	0.339853	0.136292
1	0	1	1432274.804	397923.0256	3.599377	0.511077	0.141991
1	0	2	1890969.357	426340.2903	4.435352	0.674753	0.152131
1	0	3	2309404.887	475983.5879	4.851858	0.824063	0.169845
1	0	5	3092499.495	614067.4423	5.036091	1.103494	0.219117
1	0	7	3883226.673	802031.4857	4.841738	1.385648	0.286188
1	0	9	4602481.719	1023109.042	4.498525	1.642299	0.365075
1	0	11	5337615.827	1286084.402	4.150284	1.904617	0.458912
1.05	0	-2	-119375.8835	459042.8222	-0.260054	-0.038637	0.148571
1.05	0	-1	393694.1154	439837.0231	0.895091	0.127421	0.142355
1.05	0	0	894583.97	436775.3688	2.048156	0.289536	0.141364
1.05	0	1	1423345.618	453226.3128	3.140474	0.460672	0.146689
1.05	0	2	1935358.725	486012.55	3.982117	0.626388	0.1573
1.05	0	3	2425313.712	526701.6036	4.604721	0.784964	0.170469
1.05	0	5	3332602.408	669272.8411	4.979438	1.078612	0.216613
1.05	0	7	4177193.336	869146.9615	4.806084	1.351968	0.281303
1.05	0	9	5084766.391	1114733.909	4.561417	1.645708	0.360789
1.05	0	11	5941995.791	1421887.168	4.17895	1.923154	0.4602
1.3	0	-2	-776889.8399	645890.2417	-1.20282	-0.164034	0.136374
1.3	0	-1	-101981.257	614665.3505	-0.165913	-0.021532	0.129781
1.3	0	0	492482.6982	609206.4249	0.8084	0.103984	0.128629
1.3	0	1	1126157.118	623543.4744	1.80606	0.237778	0.131656
1.3	0	2	1762037.589	655900.9621	2.686438	0.372039	0.138488
1.3	0	3	2409302.026	710620.2275	3.390421	0.508704	0.150041
1.3	0	5	3633037.331	875284.1246	4.150695	0.767085	0.184809
1.3	0	7	4995147.826	1143446.378	4.368502	1.054683	0.241429
1.3	0	9	6146283.085	1470890.33	4.178614	1.297735	0.310566
1.3	0	11	7334468.333	1890126.252	3.880412	1.548611	0.399084
1.6	0	-2	-1092851.051	911374.9635	-1.199123	-0.152329	0.127033

1.6	0	-1	-491404.4947	877563.4373	-0.559965	-0.068495	0.12232
1.6	0	0	256883.7169	862289.9938	0.297909	0.035806	0.120191
1.6	0	1	1062365.893	875293.1899	1.213726	0.148079	0.122004
1.6	0	2	1782417.109	909493.875	1.95979	0.248445	0.126771
1.6	0	3	2536193.863	969254.1364	2.616645	0.353511	0.135101
1.6	0	5	3960207.171	1155987.588	3.425822	0.551999	0.161129
1.6	0	7	5294201.442	1431540.988	3.698253	0.73794	0.199537
1.6	0	9	6781368.678	1818651.527	3.728789	0.94523	0.253495
1.6	0	11	8238776.214	2305182.589	3.574023	1.148373	0.321311
2	0	-2	-1526097.224	1335687.072	-1.142556	-0.136139	0.119153
2	0	-1	-597967.1681	1282925.883	-0.466096	-0.053343	0.114446
2	0	0	288151.6339	1260012.248	0.22869	0.025705	0.112402
2	0	1	1207910.37	1265546.06	0.954458	0.107754	0.112896
2	0	2	2353351.512	1313996.577	1.790988	0.209936	0.117218
2	0	3	3200868.339	1385971.128	2.309477	0.285541	0.123639
2	0	5	4951979.582	1623994.167	3.049259	0.441753	0.144872
2	0	7	6989212.612	2087204.946	3.348599	0.623489	0.186194
2	0	9	9166916.426	2670038.407	3.433253	0.817756	0.238187
2	0	11	10100796	3024097.29	3.340103	0.901065	0.269771

D.4 Fine Mesh, Refinement

Mach	Beta	AOA	Lift	Drag	L/D	CL	CD
0.1	0	-2	2440.895593	1210.87505	2.015811	0.087098	0.043208
0.1	0	-1	5601.283349	1208.761098	4.633904	0.19987	0.043132
0.1	0	0	8761.444514	1274.143155	6.876342	0.312634	0.045465
0.1	0	1	11924.50725	1393.17678	8.55922	0.425501	0.049713
0.1	0	2	15097.16645	1571.69325	9.60567	0.538711	0.056083
0.1	0	3	18288.15831	1805.959055	10.126563	0.652575	0.064442
0.1	0	5	24663.75583	2457.977222	10.034168	0.880075	0.087708
0.1	0	7	31010.01286	3366.906874	9.210238	1.106528	0.120141
0.1	0	9	37334.68037	4569.309188	8.170749	1.33221	0.163046
0.1	0	11	43611.05974	6093.423092	7.157071	1.55617	0.217431
0.3	0	-2	23228.94292	11003.20093	2.111108	0.092097	0.043625
0.3	0	-1	52179.52687	11016.63352	4.736431	0.20688	0.043678
0.3	0	0	81017.48821	11591.55161	6.989357	0.321216	0.045958
0.3	0	1	109923.6134	12774.4493	8.604959	0.435822	0.050648
0.3	0	2	138793.0071	14354.75772	9.668781	0.550282	0.056913
0.3	0	3	168012.7882	16607.48386	10.116691	0.666132	0.065845
0.3	0	5	226255.194	22668.78746	9.980913	0.897049	0.089876
0.3	0	7	284236.1775	31064.85858	9.149766	1.12693	0.123165
0.3	0	9	342019.7461	42074.36336	8.128935	1.356029	0.166815
0.3	0	11	398870.6166	56143.99074	7.104422	1.581429	0.222598
0.7	0	-2	166759.2685	77567.45834	2.149861	0.121438	0.056486
0.7	0	-1	338001.6699	75621.17958	4.469669	0.24614	0.055069
0.7	0	0	509675.9536	82551.39315	6.174044	0.371158	0.060116
0.7	0	1	684878.4696	83735.01251	8.179117	0.498744	0.060978
0.7	0	2	839469.1015	102050.8796	8.225986	0.61132	0.074316
0.7	0	3	957670.8899	98183.20455	9.753918	0.697398	0.071499
0.7	0	5	1350111.092	155164.0057	8.701187	0.983182	0.112994
0.7	0	7	1723234.011	213558.0643	8.069159	1.254898	0.155518
0.7	0	9	2011258.393	274949.1827	7.315019	1.464644	0.200224
0.7	0	11	2388845.375	383343.5207	6.231605	1.739611	0.279159
0.9	0	-2	608535.6382	204502.3746	2.97569	0.268078	0.090089
0.9	0	-1	957162.4897	207082.595	4.622129	0.421659	0.091226
0.9	0	0	1289762.923	222261.5926	5.802905	0.568179	0.097913
0.9	0	1	1591320.221	247916.7387	6.418769	0.701024	0.109215
0.9	0	2	1909887.922	289528.5855	6.596544	0.841362	0.127546
0.9	0	3	2218938.166	339815.2477	6.529837	0.977508	0.149699
0.9	0	5	2755167.725	444943.1132	6.19218	1.213733	0.196011
0.9	0	7	3355026.344	597848.4537	5.611834	1.477989	0.26337
0.9	0	9	3900905.628	775163.4405	5.032365	1.718465	0.341483
0.9	0	11	4454174.057	995534.4887	4.474153	1.962196	0.438563
0.95	0	-2	266620.6416	290853.51	0.916684	0.105416	0.114997
0.95	0	-1	672474.6897	281100.058	2.392297	0.265882	0.111141
0.95	0	0	1072395.519	285950.5297	3.750283	0.424002	0.113059
0.95	0	1	1473743.009	303410.4426	4.857259	0.582686	0.119962

0.95	0	2	1852249.063	336574.3184	5.503239	0.732339	0.133074
0.95	0	3	2202436.701	384912.6354	5.721913	0.870796	0.152186
0.95	0	5	2888058.542	511631.9534	5.644797	1.141876	0.202288
0.95	0	7	3548184.482	675320.1332	5.254078	1.402876	0.267007
0.95	0	9	4224370.903	882346.5328	4.787655	1.670226	0.348861
0.95	0	11	4878792.142	1132449.997	4.308174	1.92897	0.447746
0.99	0	-2	-16579.92961	373584.1464	-0.044381	-0.006036	0.136012
0.99	0	-1	507192.055	339887.7335	1.492234	0.184656	0.123744
0.99	0	0	966800.7244	345743.4526	2.796295	0.351987	0.125876
0.99	0	1	1422297.192	360596.3494	3.944292	0.517822	0.131284
0.99	0	2	1871218.916	390510.757	4.791722	0.681263	0.142175
0.99	0	3	2281284.107	432580.9825	5.273658	0.830557	0.157492
0.99	0	5	3095452.787	564419.9331	5.484308	1.126974	0.205491
0.99	0	7	3863934.191	742415.143	5.204547	1.406759	0.270294
0.99	0	9	4611748.222	966761.4512	4.770306	1.679019	0.351973
0.99	0	11	5321178.603	1236487.588	4.303463	1.937304	0.450173
1	0	-2	39543185.01	-1316428.983	0.474958404	0.0648	0.136433
1	0	-1	473484.9222	361083.3675	1.31129	0.168953	0.128845
1	0	0	950335.2751	362440.7283	2.622043	0.339107	0.129329
1	0	1	1418642.427	375679.282	3.776206	0.506213	0.134053
1	0	2	1880463.254	406334.0707	4.627875	0.671004	0.144992
1	0	3	2322298.526	446630.9919	5.199591	0.828664	0.159371
1	0	5	3128068.456	582815.5952	5.367167	1.116186	0.207966
1	0	7	3932334.805	758699.1133	5.182996	1.403172	0.270726
1	0	9	4664420.264	990646.3317	4.708462	1.664401	0.353491
1	0	11	5410352.95	1267184.154	4.269587	1.930571	0.452168
1.05	0	-2	-129731.0141	428032.6465	-0.303087	-0.041988	0.138535
1.05	0	-1	369220.974	414277.4441	0.891241	0.1195	0.134083
1.05	0	0	864930.6315	414169.5853	2.088349	0.279939	0.134048
1.05	0	1	1352593.033	427368.5986	3.164933	0.437773	0.13832
1.05	0	2	1873774.635	460366.6208	4.070179	0.606456	0.149
1.05	0	3	2389623.504	508006.2346	4.703926	0.773412	0.164419
1.05	0	5	3334026.649	650453.633	5.125695	1.079073	0.210522
1.05	0	7	4177653.204	827635.8778	5.047695	1.352116	0.267868
1.05	0	9	5068660.555	1087283.657	4.661765	1.640495	0.351904
1.05	0	11	5936534.458	1396474.8	4.251086	1.921386	0.451975
1.3	0	-2	-709329.099	626038.9734	-1.133043	-0.149769	0.132183
1.3	0	-1	-164437.0269	604204.4658	-0.272155	-0.034719	0.127573
1.3	0	0	541397.1352	594532.9978	0.910626	0.114311	0.125531
1.3	0	1	1115699.24	601350.4823	1.855323	0.23557	0.12697
1.3	0	2	1780368.117	641633.1909	2.774744	0.37591	0.135475
1.3	0	3	2396405.782	694774.3515	3.449186	0.505981	0.146696
1.3	0	5	3505242.443	847593.0706	4.135525	0.740102	0.178962
1.3	0	7	4862483.46	1108223.886	4.387636	1.026672	0.233992
1.3	0	9	5463169.813	1380380.659	3.957727	1.153502	0.291456
1.3	0	11	7346367.346	1877363.618	3.91313	1.551123	0.396389
1.6	0	-2	-1078478.723	903103.1435	-1.194192	-0.150325	0.12588

1.6	0	-1	-433829.4312	864485.6252	-0.501835	-0.06047	0.120498
1.6	0	0	224227.3418	849373.7221	0.263991	0.031254	0.118391
1.6	0	1	1075525.495	856977.7416	1.255022	0.149914	0.119451
1.6	0	2	1813895.842	898599.8786	2.01858	0.252832	0.125253
1.6	0	3	2541379.545	961251.6382	2.643823	0.354234	0.133985
1.6	0	5	3885476.101	1142494.709	3.40087	0.541582	0.159248
1.6	0	7	5363803.447	1427283.406	3.758051	0.747641	0.198944
1.6	0	9	6945417.792	1828511.418	3.7984	0.968097	0.25487
1.6	0	11	8244595.029	2305136.884	3.576618	1.149184	0.321305
2	0	-2	-1242338.532	1306833.377	-0.950648	-0.110826	0.116579
2	0	-1	-1721674.369	1407950.704	-1.222823	-0.153586	0.125599
2	0	0	-233322.8729	1266049.923	-0.184292	-0.020814	0.112941
2	0	1	973150.0578	1267710.283	0.767644	0.086812	0.113089
2	0	2	2266740.761	1298600.906	1.745525	0.20221	0.115845
2	0	3	2388334.654	1468217.762	1.62669	0.213057	0.130976
2	0	5	4679204.259	1704064.014	2.745909	0.417419	0.152015
2	0	7	6611456.117	1979054.411	3.340715	0.58979	0.176546
2	0	9	8910039.247	2570785.048	3.465883	0.79484	0.229333
2	0	11	11237085.84	3321499.638	3.383136	1.00243	0.296302

D.5 Viscous Case

Mach	Beta	AOA	Lift	Drag	L/D	CL	CD
0.3	0	-6	-93850.64749	19765.90646	-4.748107	-0.372096	0.078367
0.3	0	-3	-7341.596136	15153.36418	-0.484486	-0.029108	0.06008
0.3	0	-2	21256.83526	14622.85663	1.453672	0.084278	0.057976
0.3	0	-1.5	35510.37261	14547.70875	2.44096	0.14079	0.057678
0.3	0	-1	49738.77906	14689.09039	3.386103	0.197203	0.058239
0.3	0	-0.7	58181.27314	14884.76751	3.908779	0.230675	0.059015
0.3	0	-0.5	63950.94586	14881.17257	4.29744	0.253551	0.059
0.3	0	-0.4	66856.91774	14902.90789	4.486166	0.265072	0.059087
0.3	0	-0.3	69738.93824	15161.90243	4.599617	0.276499	0.060113
0.3	0	-0.2	72540.36999	15183.83208	4.777474	0.287606	0.0602
0.3	0	-0.1	75458.39481	15143.93038	4.982748	0.299175	0.060042
0.3	0	0	78248.2569	15104.9125	5.180318	0.310236	0.059887
0.3	0	0.1	81120.66563	15375.5118	5.275965	0.321625	0.06096
0.3	0	0.2	83830.53098	15251.69108	5.496474	0.332369	0.060469
0.3	0	0.3	86788.26666	15491.84434	5.602191	0.344095	0.061422
0.3	0	0.4	89666.37207	15620.97287	5.740127	0.355506	0.061934
0.3	0	0.5	92554.1527	15790.33942	5.861442	0.366956	0.062605
0.3	0	0.7	98369.65251	16006.66711	6.145542	0.390013	0.063463
0.3	0	1	106961.6708	16459.98838	6.498283	0.424078	0.06526
0.3	0	1.5	121254.7743	16876.65235	7.184765	0.480747	0.066912
0.3	0	2	135781.395	18008.58123	7.539816	0.538342	0.0714
0.3	0	3	164852.6468	20350.27612	8.100757	0.653602	0.080684
0.3	0	6	251569.7614	30528.10339	8.240596	0.997416	0.121037
0.3	0	9	339891.5756	44899.98083	7.569972	1.347591	0.178018
0.3	0	12	428264.843	65302.82681	6.558136	1.697971	0.258911

E Images

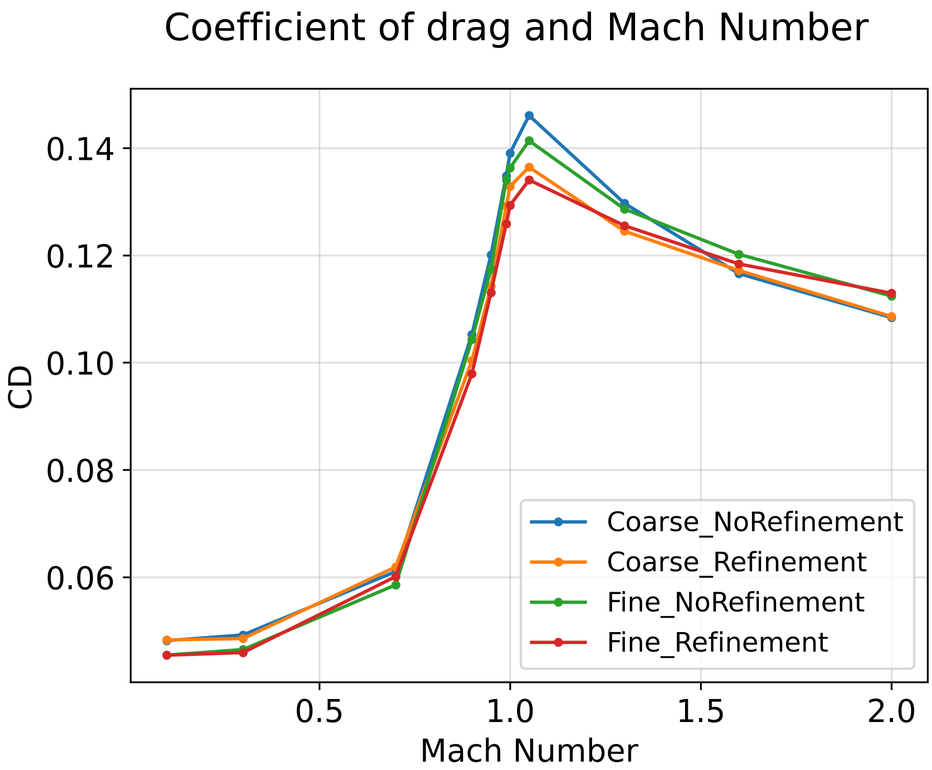


Figure 10: Relationship between CD and Mach

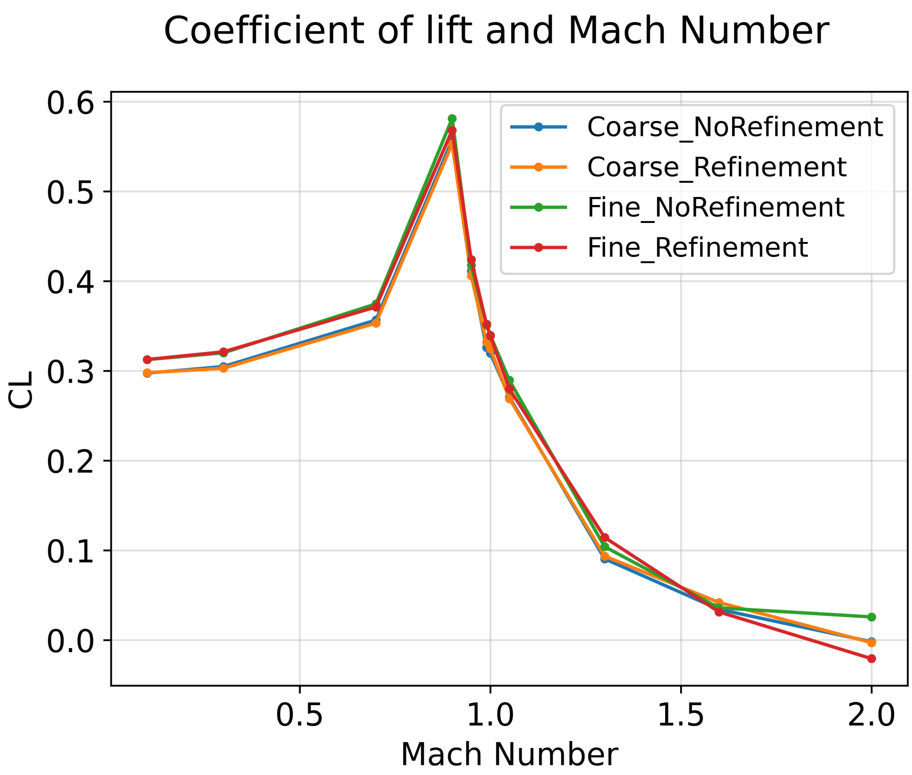


Figure 11: Relationship between CL and Mach

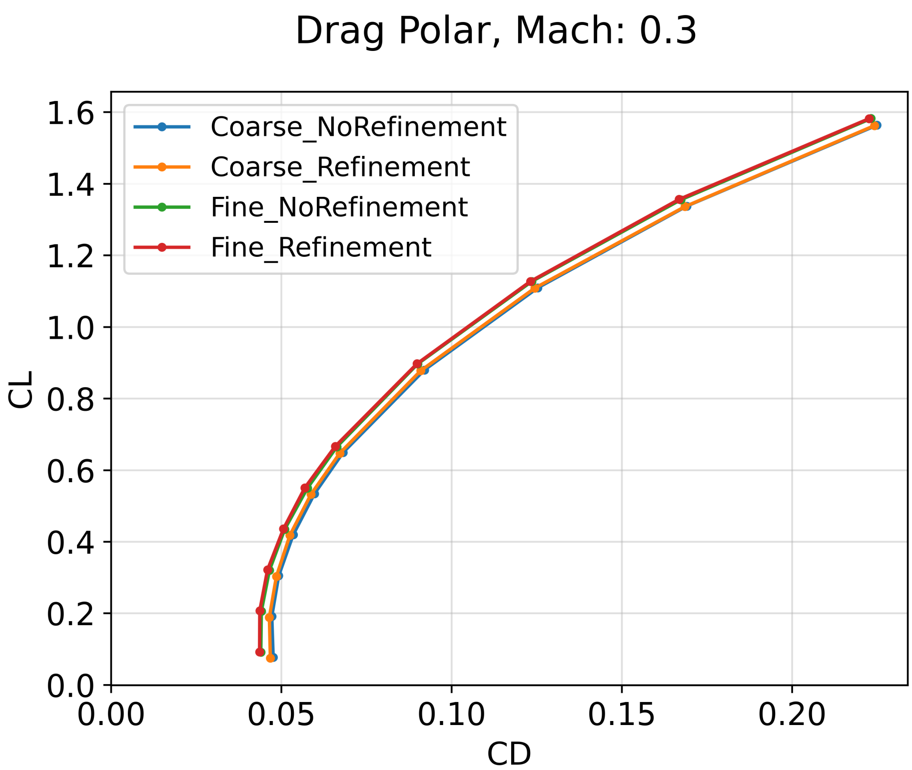


Figure 12: Drag polar at M0.3

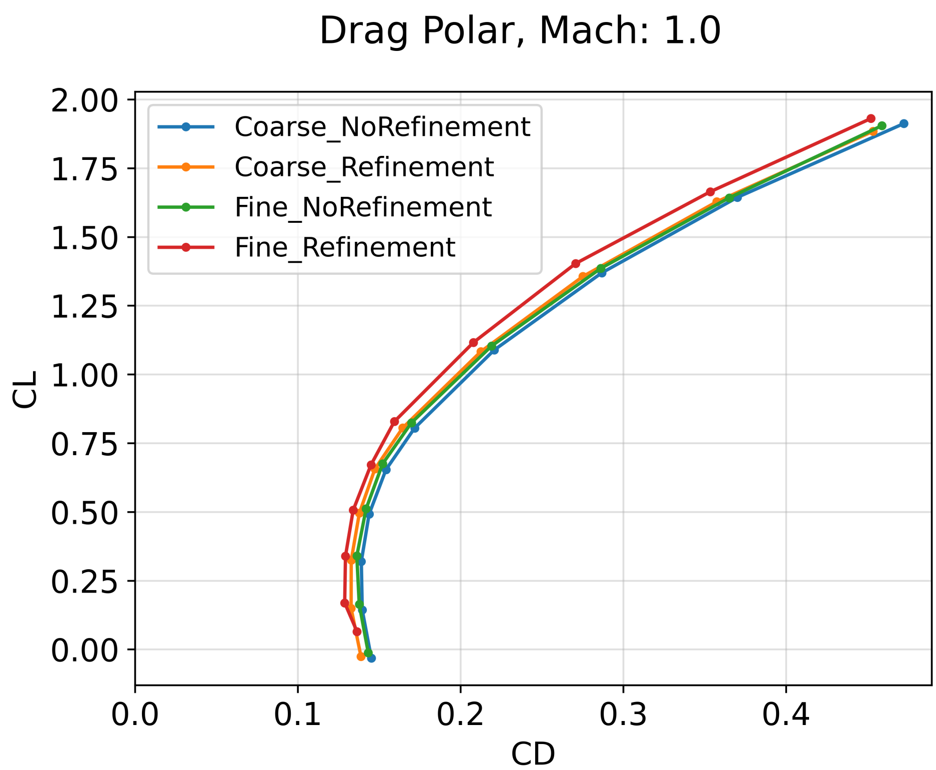


Figure 13: Drag polar at M1

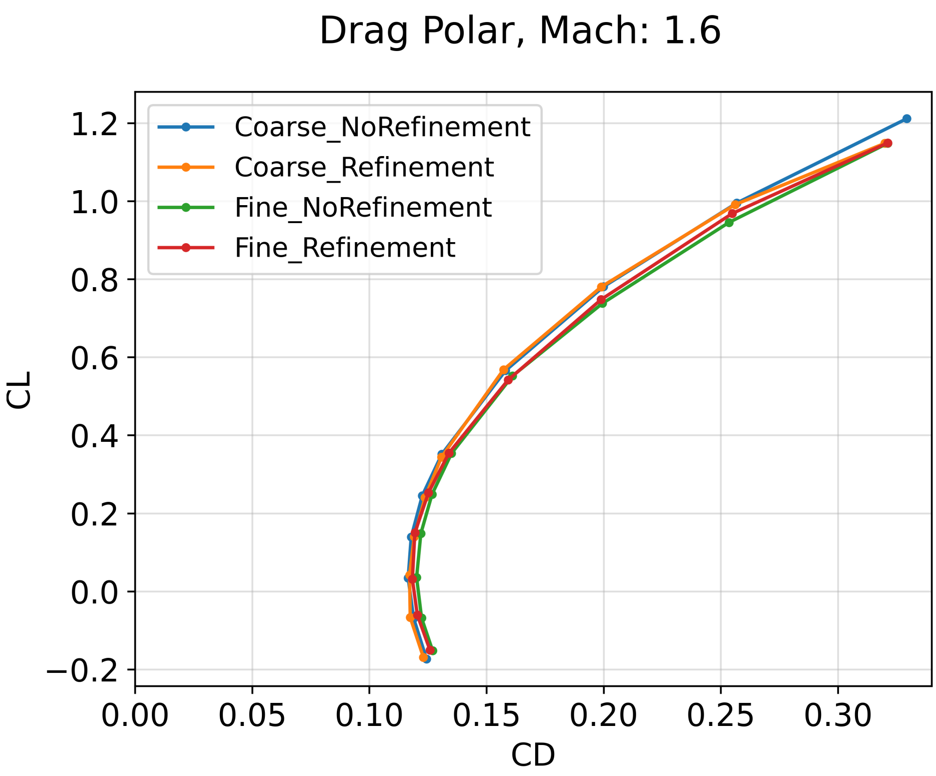


Figure 14: Drag polar at M1.6

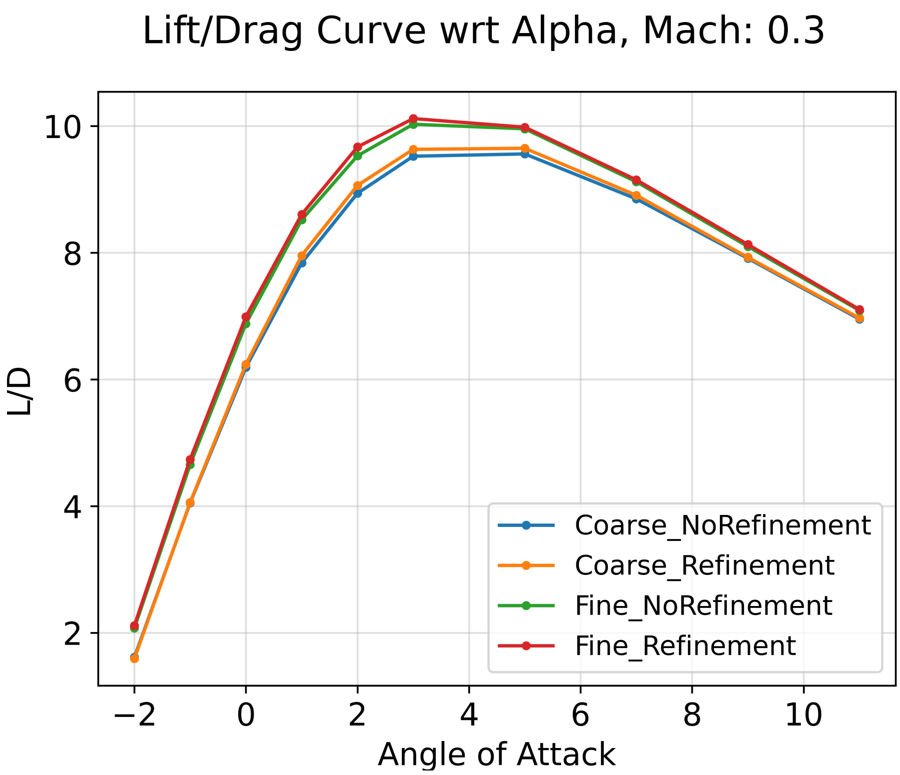


Figure 15: Relationship between L/D and alpha at M0.3

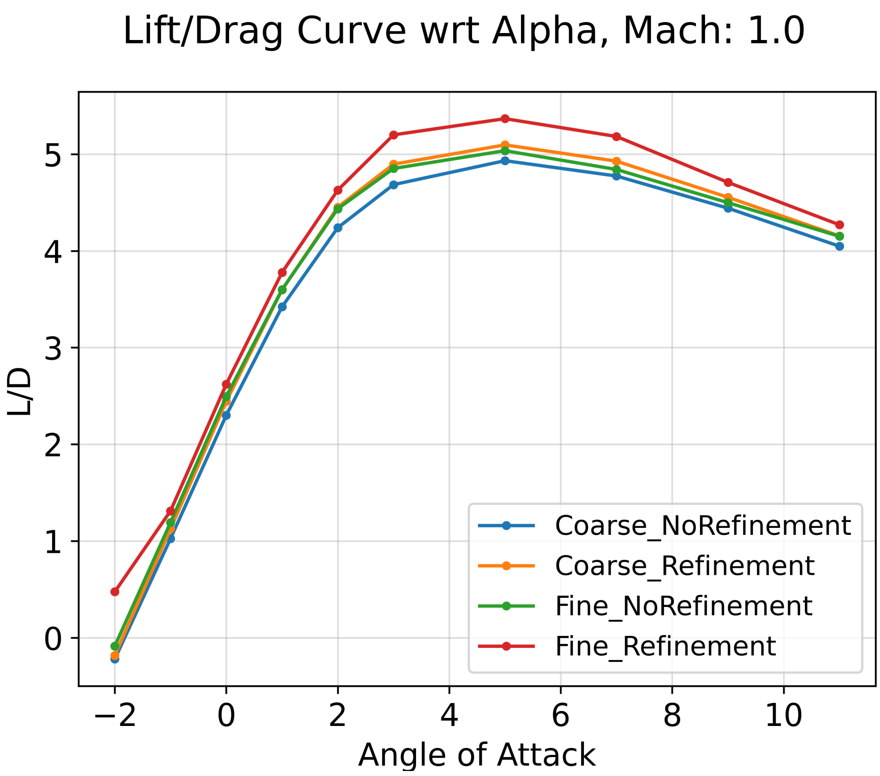


Figure 16: Relationship between L/D and alpha at M1

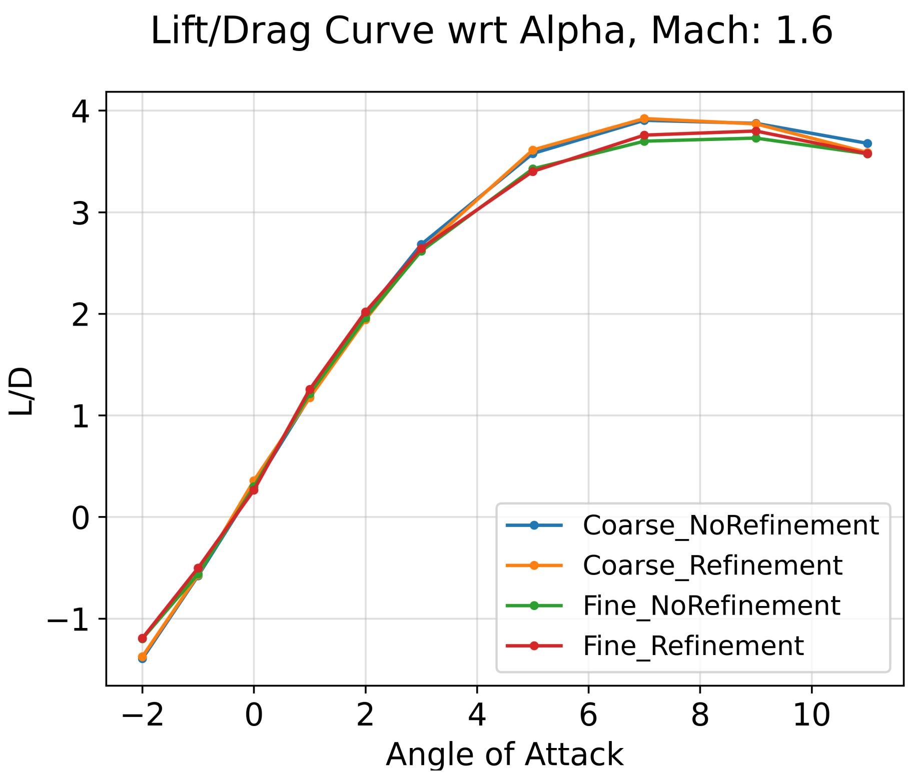


Figure 17: Relationship between L/D and alpha at M1.6

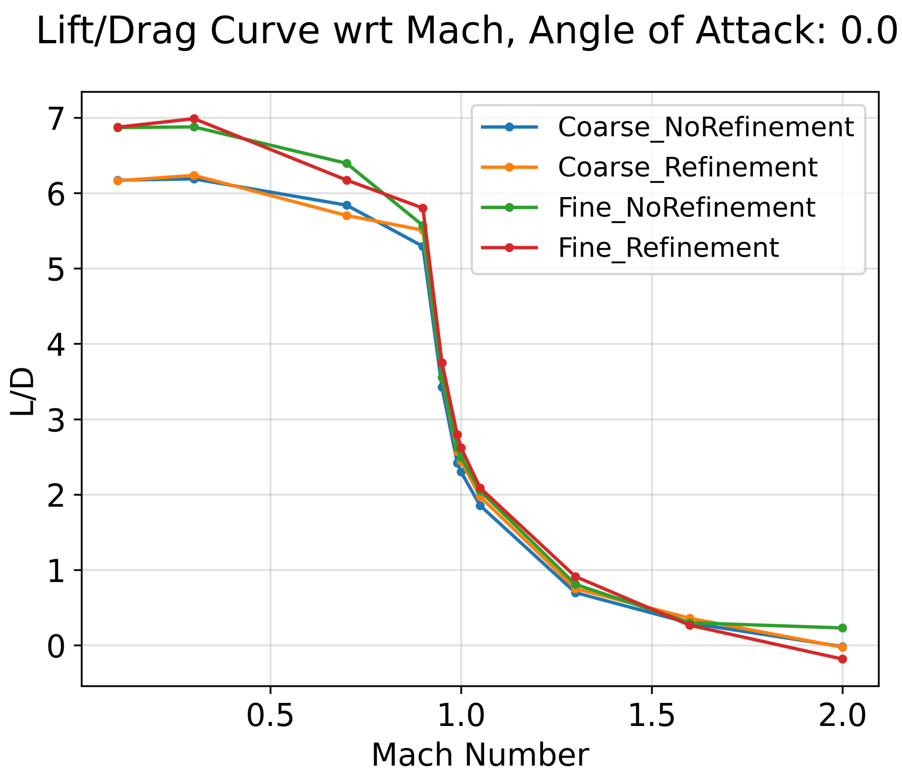


Figure 18: Relationship between L/D and Mach number, alpha = 0

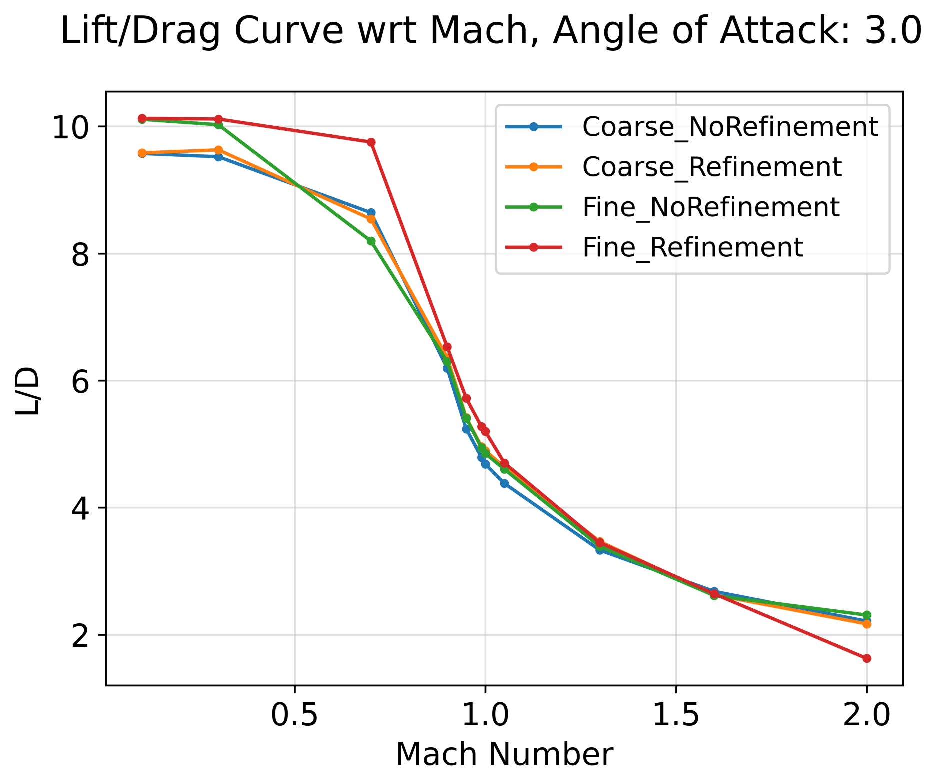


Figure 19: Relationship between L/D and Mach number, alpha = 3

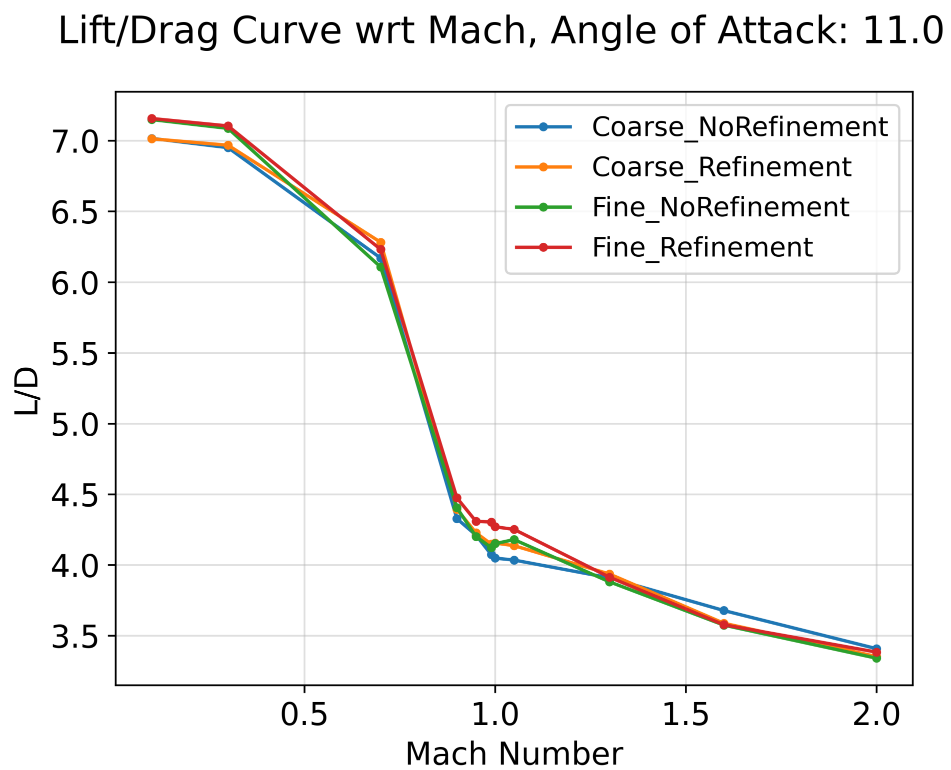


Figure 20: Relationship between L/D and Mach number, alpha = 11

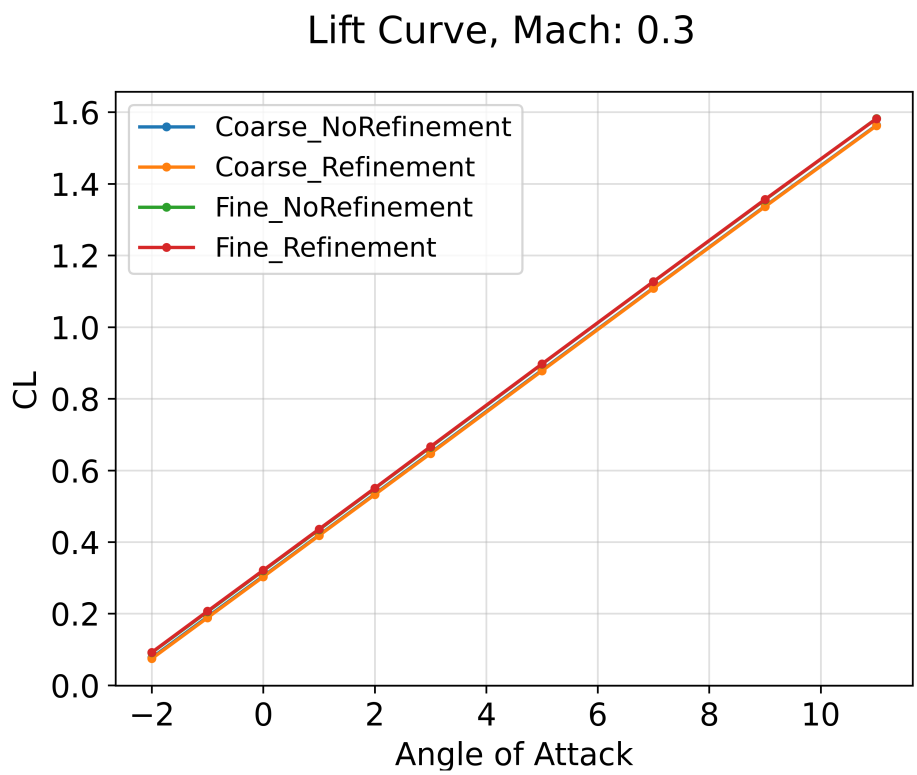


Figure 21: Lift curve at Mach 0.3

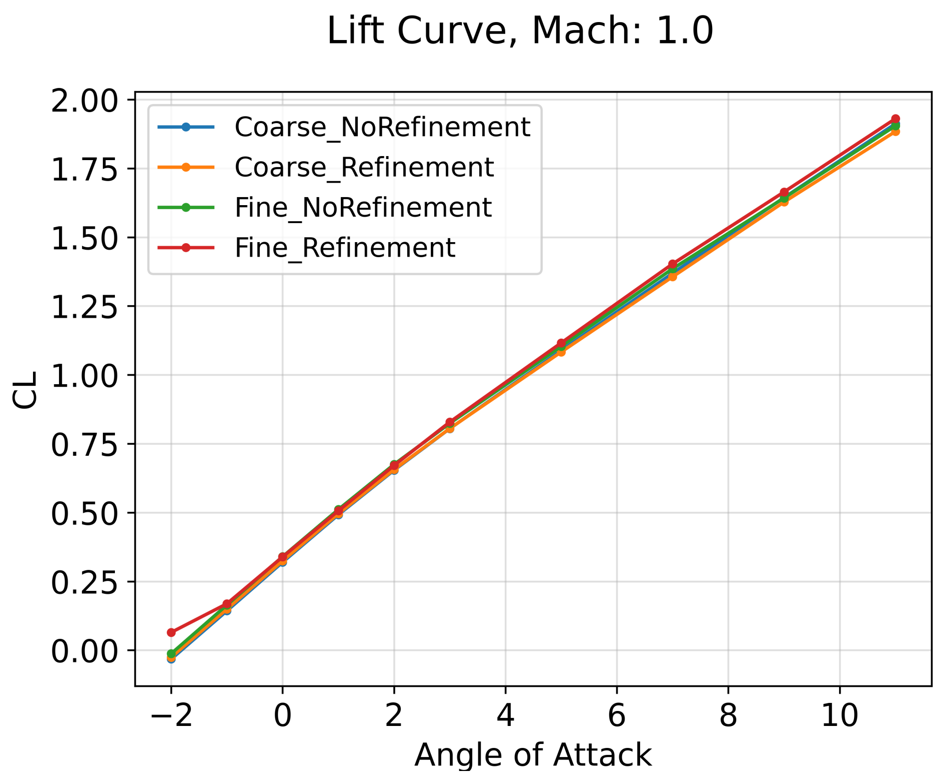


Figure 22: Lift curve at Mach 1

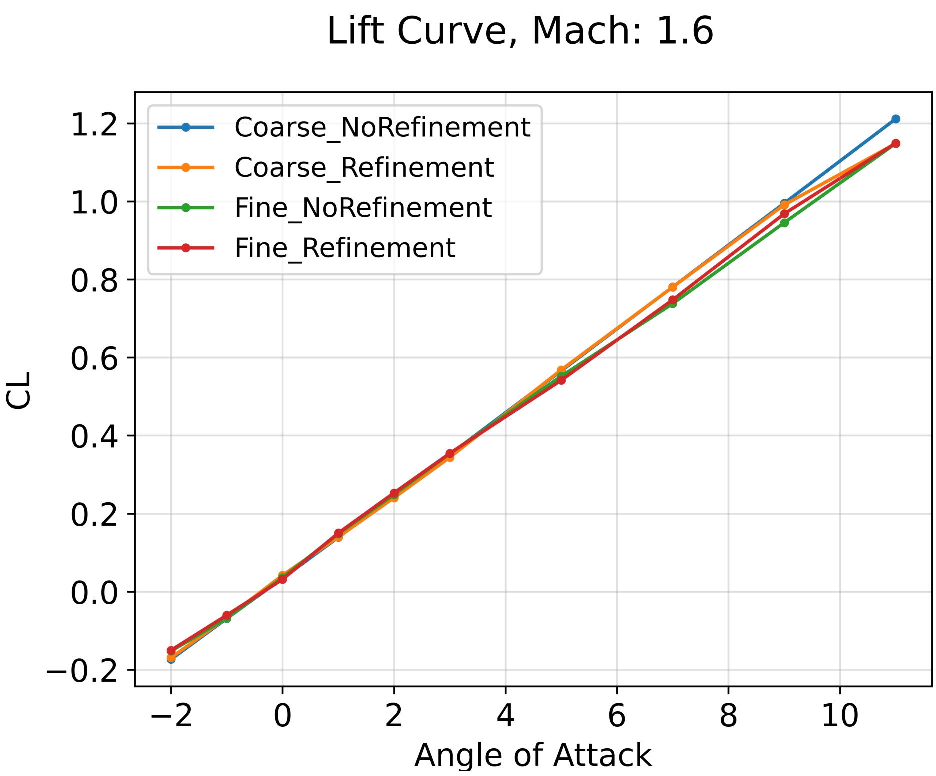


Figure 23: Lift curve at Mach 1.6