

Parameter Estimation of Fundamental Technical Aircraft Information Applied to Aircraft Performance

Mike Vallone ^{*}, Robert A. McDonald [†]

California Polytechnic State University, San Luis Obispo, CA, 93405, USA

Inverse problems can be applied to aircraft in many areas. One of the disciplines within the aerospace industry with the most openly published data is in the area of aircraft performance. Many aircraft manufacturers publish performance claims, flight manuals and Standard Aircraft Characteristic (SAC) charts without any mention of the more fundamental technical information such as C_{D0} . With accurate tools, generalized aircraft models and a few curve-fitting techniques, it is possible to evaluate vehicle performance and estimate these technical parameters.

With this goal a program has been written in Matlab to calculate the fundamental information behind a general aircraft. The current results are promising with more work underway to further improve them. The results shown are for the Northrop F-5. While the results look good, the overall accuracy of the program is only as good as the data provided. For example, the maximum speed reported by a company may assume that the aircraft is in a dive with engines at full throttle. Or, the measurements may be overly optimistic, with an important characteristic such as ground friction during takeoff left out of the picture. If this information is not reported then an accurate analysis is not possible.

Nomenclature

\mathcal{R}	Aspect Ratio
C_L	Lift Coefficient
$C_{L_{max}}$	Maximum Lift Coefficient
C_D	Drag Coefficient
C_{D0}	Zero-lift Drag Coefficient
D	Drag, lbf
e	Oswald Efficiency Factor
F	Force, lbf
g	Gravitational Constant, ft/sec ²
J	Jacobian
L	Lift, lbf
M	Mach Number
nm	Nautical Miles
P	Pressure, psf
Re	Reynolds Number
T	Thrust, lbf
$Temp$	Temperature, degrees Fahrenheit
V	Velocity, ft/sec
W	Weight, lbf
δ_0	Non-dimensional Pressure
θ_0	Non-dimensional Temperature
<i>Subscript</i>	
i	Variable number
x	Parallel to Freestream Velocity
z	Perpendicular to Freestream Velocity
∞	Free Stream

^{*}One Grand Avenue, Student Member

[†]Lockheed Martin Assistant Professor, Aerospace Engineering, One Grand Avenue, AIAA Member

I. Introduction

This research is intended to discover general technical values, such as C_{D_0} and $C_{L_{max}}$, along with a complete drag polar and engine deck corresponding to the plane in question. It is also meant to extend the knowledge of the authors in the areas of inverse design and aircraft performance. As will be shown, relatively few assumptions are necessary to build complex profiles with which full aircraft dynamics can be modeled.

A program with this capability can be useful in many areas of the aircraft industry, ranging from academia to conceptual design at premier aircraft companies. Design teachers nationwide could use this to extract historical information from basic flight data for use in preliminary aircraft design. Practical validation of students' final design projects and models could be completed with the help of this program. Public and private companies could utilize this program to understand the effects of changing key aircraft components. As an example, imagine a company wishes to modify a Cessna Citation X to catch drug runners smuggling drugs into this country. With a maximum speed of Mach 0.92 and a range of about 3,000 nm¹, this plane is well suited for the task. However, thousands of pounds of equipment might need to be added to this plane, which could drastically diminish the performance expected from this plane. Swapping the Rolls-Royce engines with more powerful ones might deliver the necessary power to keep this plane competitive. Knowing the specific drag information about the original aircraft, which can be obtained with this program, will prove crucial in estimating the new performance characteristics of the heavily modified plane.

One crucial part of this research involves the assumptions. Each performance metric evaluated requires additional and different assumptions. For example, rate of climb requires only assuming general engine and drag models. However, takeoff analysis involves a few more, including ground roll coefficient, C_L on the ground, and obstacle height. Luckily many of these assumptions can be reasonably estimated or are given; there are standard values to use for ground roll coefficient and obstacle height is dictated by the Federal Aviation Regulations (FARs)². Most of these assumptions can be changed with ease, with the goal to have all assumptions seen by the user and easy to change should the occasion arise where this is necessary.

At this stage the research is still ongoing. This is a work in progress coupled with the M.S. thesis project of the author, to be defended in the coming year. At this point the supersonic drag polar and afterburning-turbojet models have been validated along with the takeoff and rate of climb subroutines. The goal of this paper is to show the promise of this research with a specific test case, the Northrop F-5, and to extend the research to multiple aircraft and subroutines for the thesis work.

II. Background

Three main areas were explored in the writing of this program. First and foremost was discovering the methods and theories behind Inverse Problems. This accounted for a large portion of the research as this was the portion the author was least familiar with. The other large chunk involved development of the Aircraft Performance routines used to generate the computer calculated results. The third and final section consisted of finding generalized models for drag and thrust with parameters to vary.

A. Inverse Problems

An inverse problem is the task of using given data to deduce model parameters. This can be thought of simply as reverse-engineering. While this can be a straightforward task, most non-linear problems are ill-posed or ill-conditioned, resulting in many difficulties along the way³. Returning to the physics and dynamics behind the equations themselves later, a function G may be specified such that m and d are related by

$$G(m) = d. \quad (1)$$

In practice, m represents some unknown model parameters, and d represents observations in time and space or a set of discrete points. We will only be looking at the discrete data case here as the goal is to analyze aircraft performance data, supplied as discrete points. It is important to note that this problem is made harder by the fact that there is inherent noise in most data, and this error must be dealt with. Error will be discussed in more detail later.

Forward problems involve using m to find d . This is the process usually employed during the aircraft design process, but as we are reverse engineering airplanes, we are searching for these precious parameters m . G can be anything from a simple equation to something much more complex, such as ordinary differential equations or iterative processes that have no analytical equation.

As is the case in most mathematical proceedings, linear inverse problems are considerably easier to solve than their non-linear counterparts. It can be shown that in the case of the linear inverse problem (1) can always be written in the form of a linear system of algebraic equations⁴

$$\mathbf{Gm} = \mathbf{d}. \quad (2)$$

While this is very useful for some applications, the performance section will go into detail on the models themselves, making it quite evident that this is not a linear problem.

Due to the non-linear nature of this problem there are two general methods that will be investigated to obtain a solution⁴. These are to use either a non-linear curve fitting technique or Occam's Inversion. The theory behind both methods will be discussed here; at this time, only the curve fitting technique has been implemented.

1. Non-Linear Curve Fitting

Non-linear curve fitting is implemented by using a non-linear optimizer on problems of the form⁵

$$\min \|f(x)\|_2^2 = \min ((d_1 - G_1(m))^2 + (d_2 - G_2(m))^2 + \dots + (d_n - G_n(m))^2). \quad (3)$$

Matlab's function lsqnonlin from the optimization toolbox is currently being used as the non-linear curve fitting tool. The algorithm used is Levenberg-Marquardt, which is primarily used for curve fitting problems such as this one⁶. The algorithm minimizes

$$S(\beta) = \sum_{i=1}^m [y_i - f(x_i, \beta)]^2. \quad (4)$$

This routine combines the Gauss-Newton technique with that of steepest descent. The Gauss-Newton method employs the numerically approximated first and second derivatives to select the next search direction, while the steepest descent follows the negative of the gradient to the minimum of a function. Each method is better applied in different circumstances which is capitalized on by the Levenberg-Marquardt routine. The values of x are updated by

$$x_{i+1} = x_i - (H + \lambda \text{diag}[H]^{-1})^{-1} \nabla f(x_i) \quad (5)$$

where the Hessian, H , is approximated numerically. As with most optimizers this is a strictly numerical process, and requires an initial guess, β above. Some problems can use a random initial guess, while other more sensitive problems require more finesse⁶. One additional long term goal of this research is to determine the sensitivity to the initial guess vector.

2. Occam's Inversion

Occam's Inversion techniques have not yet been researched enough to have an implementation in Matlab. However, the development of an algorithm to implement this technique will be key to finding the optimal way to solve this problem. According to William of Ockham, who this technique is named after, simpler explanations should always be preferred to more complex ones⁴. This rule has since become known as Occam's razor. Occam's inversion minimizes $\|\mathbf{Lm}\|_2$ subject to the constraint $\|\mathbf{G(m)} - \mathbf{d}\| \leq \delta$. The algorithm should be easy to implement⁴ and, after beginning with an initial solution \mathbf{m}^0 , goes as follows:

1. $\mathbf{m}^{k+1} = (\mathbf{J}(\mathbf{m}^k)^T \mathbf{J}(\mathbf{m}^k) + \alpha^2 \mathbf{L}^T \mathbf{L})^{-1} \mathbf{J}(\mathbf{m}^k)^T \mathbf{d}(\mathbf{m}^k)$
2. Select largest α such that $\chi^2(\mathbf{m}^{k+1}) \leq \delta^2$
 - 2.5 If no such value occurs, select α that minimizes $\chi^2(\mathbf{m}^k + 1)$
3. Iterate. Stop when the sequence converges to $\chi^2 = \delta^2$

\mathbf{L} is a roughening matrix involved in the regularization of the non-linear least squares problem. For zeroth-order, \mathbf{L} is equal \mathbf{I} , or it can be set to the finite-difference approximations of the first or second derivatives for higher order methods⁴. More research will be done to implement a working version of this algorithm. As it stands now it is an interesting concept, and will be explored further in the upcoming months.

B. Aircraft Performance Models

In order to solve the inverse problems associated with aircraft performance some work must be done to solve the forward problem first. Multiple aircraft performance metrics were looked at and MATLAB was used to implement code to calculate them. The performance metrics below were chosen for this study due to the data typically available. The assumptions used to evaluate these performance metrics were chosen to coincide with the Standard Aircraft Characteristics (SAC) charts; however, these assumptions can easily be changed to fit the data that is provided.

1. Rate of Climb

The first metric examined was rate of climb (RoC). This was done first due to the relative simplicity compared to future performance metrics. Looking at an aircraft climbing at an exaggerated climb angle and summing the forces, it is simple to derive the equation for RoC , with the result shown here⁷.

$$RoC = \frac{(T - D)V}{W} \quad (6)$$

Rate of climb can be used to find many aircraft characteristics. The two found thus far using this performance metric are maximum *RoC* and maximum velocity. Maximum *RoC* is found by minimizing the negative of equation (C.), found using Matlab's canned optimizer *fminbnd*. Maximum velocity was found by minimizing the square of equation (C.), utilizing the same optimization routine.

2. Takeoff

The second performance metric examined was the takeoff length, both the ground roll and total distance to clear an obstacle. This analysis is done by running a code written to work with MATLAB's ODE45, an ordinary differential equation solver used to analyze initial value problems of the nonstiff variety. Each section was evaluated by analyzing a two degree of freedom model and integrating the equations of motion.

C. Generalized Aircraft Models

The main goals of this inverse problem are to deduce information about an aircraft that is not readily available. This is not possible without detailed generalized models for the drag and thrust of each plane. Since it is assumed there is no previous knowledge available about either the engine deck or drag polar, models must be substituted for the originals, and the coefficients of each equation are the solutions to the inverse problem.

1. Drag Polar

The base model for the drag polar is simple. The basic subsonic format is given as

$$C_D = C_{D_0} + kC_L^2 \quad (7)$$

$$k = \frac{1}{\pi e A R} \quad (8)$$

While this works well for subsonic speeds, transonic drag rise and supersonic drag need to be handled separately. Also, C_{D_0} increases with altitude, and an approximation must be made to account for this.

Transonic and supersonic drag rise is currently approximated by a Gaussian and step function, respectively. The Gaussian is of the form

$$f(x) = A \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{(-M - \mu)^2}{2\sigma^2}. \quad (9)$$

A , σ and μ are coefficients that can be varied by the overall program to best fit the trends demonstrated in the actual drag polar. μ effects the location of the curve while σ effects the overall width of the curve. A is a simple multiplier to change the overall height of this curve. The step function modeling the supersonic drag rise takes the form

$$f(x) = A \frac{1}{1 + \exp -2(M - \mu)} \quad (10)$$

where A adjusts the height of the function and μ shifts the location of the curve.

A generic curve demonstrating this can be seen in Fig. 1. This employs values for the constants in the equations (9) and (10) close to those found for the F-5, adjusted slightly to exaggerate the transonic peak.

Transonic drag rise for subsonic aircraft is modeled with the asymptotic relationship given by⁸

$$C_{D_0} = C_{D_i} + \frac{k_1 * (M - 0.3)^2}{(k_2 - M)^n}. \quad (11)$$

According to Vinh, this equation, with n , k_1 and k_2 properly selected, gives accurate modeling of both C_{D_0} and dC_{D_0}/dM .

The value of k can also vary with both M and C_L ⁹. For this reason several both step functions and Gaussians were used to approximate the typical smooth variation of the Oswald efficiency factor. Currently a step function is used to increase k with C_L , and a Gaussian models the rise in k at approximately the speed of sound. As was found in the F-5 Substantiating Report, k increases at an approximately linear rate in the supersonic region. A step function, multiplied by Mach, was used to handle this. It is also interesting to note that in the early supersonic region C_L no longer effects k ⁹. This was also with a step function. Each of these was employed in the variation of k , and can be seen in Fig. 2 using generic coefficients.

Currently no method is employed to account for the rise in C_{D_0} with altitude. This, along with further research into transonic drag rise and variation of Oswald efficiency with Mach number, will be implemented in future work.

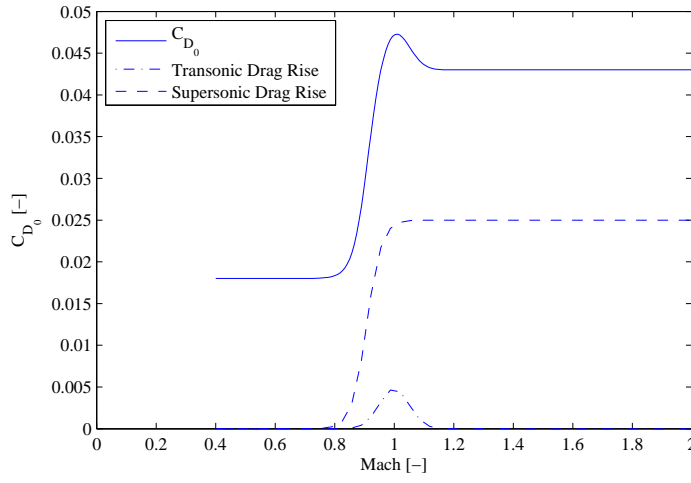


Figure 1. Generic Drag Rise Incorporating Transonic and Supersonic Corrections

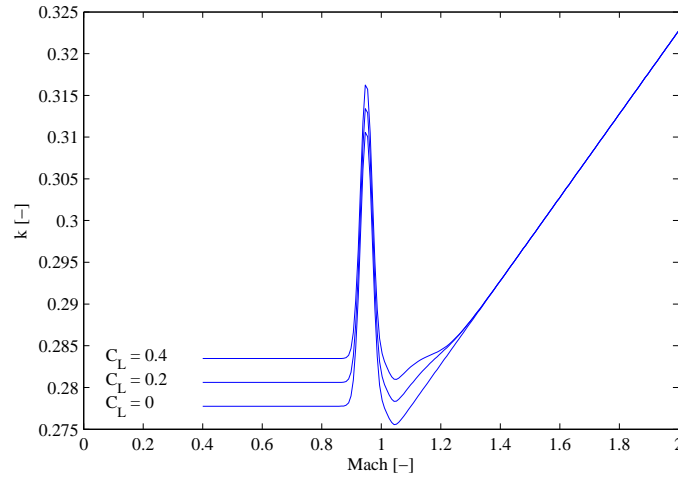


Figure 2. Generic Variation of k Incorporating Transonic and Supersonic Corrections

2. Engine Deck

There are multiple engine models that can be chosen from while running this program. This allows for flexibility in engine type, and can allow the user to switch between models to find the one with the best fit. Each model has between three and five parameters that can be varied to allow for the equation to best match the data provided. The parameters a_1 - a_5 are not related between equations and are allowed to vary freely¹⁰.

High Bypass Ratio Turbofan	$\delta_0(1 - a_1 M^{a_2} - \frac{a_3(\theta_0 - a_5)}{a_4 + M})$
Low Bypass Ratio Turbofan, with AB	$a_3 \delta_0(1 - a_1(\theta_0 - a_2)/\theta_0)$
Low Bypass Ratio Turbofan, no AB	$a_3 \delta_0(1 - a_1(\theta_0 - a_2)/\theta_0)$
Turbojet, with AB	$\delta_0(1 - a_3(\theta_0 - 1) - a_1 M^{a_2} - \frac{a_4(\theta_0 - a_5)}{\theta_0})$
Turbojet, no AB	$a_4 \delta_0(1 - a_1 M^{.5} - \frac{a_2(\theta_0 - a_5)}{(a_3 + M)\theta_0})$
Turboprop	$\delta_0(1 - a_1(M - 1)^{a_2} - \frac{a_3(\theta_0 - a_5)}{a_4(M - 0.1)})$

Table 1. Engine Deck Equations

δ_0 and θ_0 are defined as

$$\delta_0 = \frac{P}{P_{sealevel}} \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma}{\gamma - 1}}$$

$$\theta_0 = \frac{Temp}{Temp_{sealevel}} \left(1 + \frac{\gamma - 1}{2} M^2\right)$$

These current engine decks are not meant to be varied how they are here. Each of the equations above are provided with constants for each value of a_1 to a_5^{10} , and it is assumed that these coefficients can be varied freely to optimize the model for a given plane/engine combination. Whether or not this is truly the case will be investigated further and changed accordingly in future work.

D. Integration

Each of these separate topics must now be fully integrated to work together. As was described previously, Matlab's lsqnonlin handles the variables as a typical non-linear optimizer would. It calculates numerical derivatives and controls the variable selection during the whole process. These variables are passed into each performance sub-routine, which calculates it's specific performance metric. This is not possible without the engine and drag models, however, which are passed in to the performance code. The variables handled by lsqnonlin dictate the engine and drag characteristics which dictate the overall performance. Shown below in Fig. 3 is the overall construction of the code in a visual format.

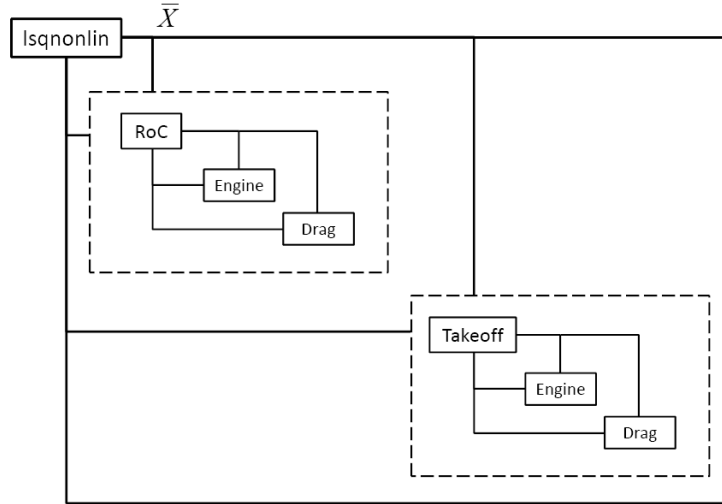


Figure 3. Overall Schematic of Code Integration

Each performance block runs independently and may call the drag and engine functions multiple times, depending on the function. There is no optimization occurring in the performance block; however, there is feedback and feed-forward in each one.

III. Code Validation

A key component of any research project is to validate any computer code written, including all components used. Generic drag polar, engine deck, rate of climb and takeoff analysis codes have been written to be used with Matlab's built in code, each of which have been validated, with the validation results shown here.

A. Drag Polar

The drag polar was built using Gaussian and step functions to simulate the actual phenomena experienced during the transonic flight regime, as no empirical relations have been found to model these occurrences. As such, it is imperative that the results of this code are compared to that of an actual aircraft. Once again the F-5 has been used as the comparison aircraft. As you can see in Fig. 4, the drag produced by the computer model matches that of the actual F-5 data.

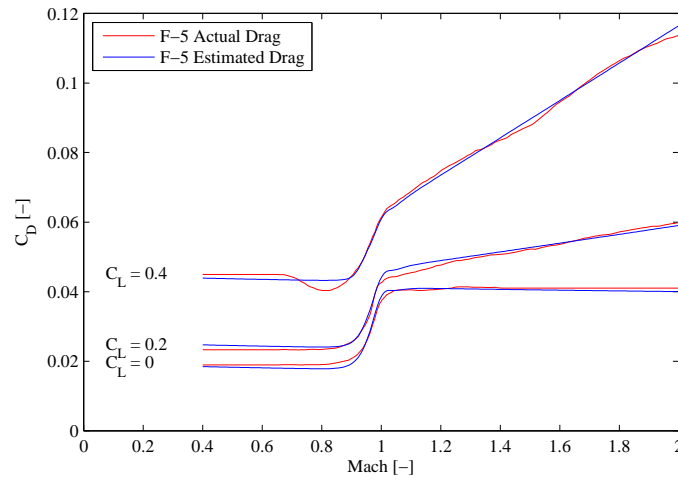


Figure 4. Drag Model Validation

B. Engine Deck

As was mentioned above, the equations shown for the engine thrust¹⁰ were not meant to have their coefficients varied as they have been here. For that reason each code must be validated against actual data to show it is being handled correctly. The Northrop F-5 has been used to validate the code written for the Turbojet engine with afterburners as there is ample performance data available for the F-5 operating with afterburners.

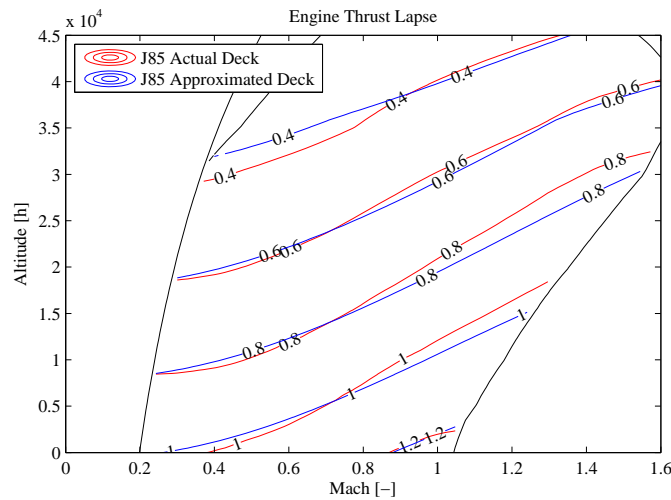


Figure 5. Engine Model Validation

It is clear from Fig. 5 that the use of these relations, with the coefficients varied as they are, is justified. The coefficients have been varied to match as best they can the actual F-5 engine data, which is fairly accurate in the flight envelope shown.

C. Rate of Climb

To validate the code written, the forward problem of calculating RoC was performed. Equation was run at each altitude from sea level to 52,000 feet. This was then compared to the values reported in the F-5 SAC chart, with the results shown in Fig. 6 below. As is evident in this figure, the results are not perfect. However, it is extremely close, with an average error of about 5%. While this error will be investigated in the future, this code is assumed to be correct as is.

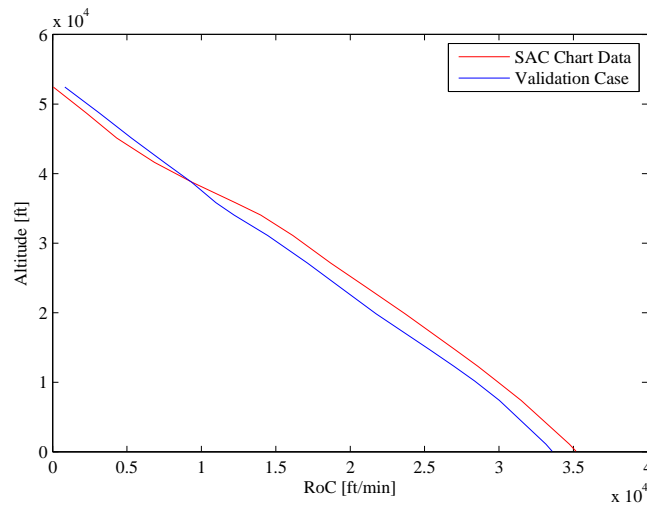


Figure 6. Rate of Climb Validation

D. Takeoff

The takeoff code is the most complex performance routine written yet, making it the most difficult to validate. The code has been validated using the engine deck and drag polar of the F-5, provided by the SAC Chart Substantiating Report⁹. Running the code at different takeoff gross weights (TOGW), it has been compared to the takeoff distances found provided⁹. There is some error, which is to be expected, as some constants are not known. Reasonable assumptions for values such as $C_{L_{gr}}$ had to be made, which will inevitably introduce error into the system.

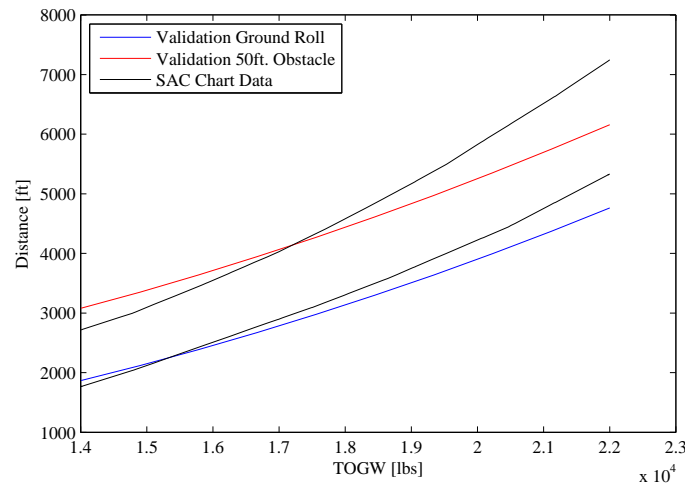


Figure 7. Takeoff Validation

The takeoff values shown in Fig. 7 are for both the computer generated takeoff values and those provided⁹. As is evident, there are some problems with this code. The primary problems that can be seen in Fig. 7 revolve around the magnitude of the variation based on changing weight. The ground roll portion does not have enough effect from the weight, as the two average slopes of the ground roll lines are not well correlated. The same problem occurs for the distance to clear an obstacle.

The main issue can be seen when comparing the distance in the air, essentially the space between the lines. In the SAC Chart provided data, the distance covered in the air approximately doubles from a weight of 14,000 lbs to 22,000 lbs. This trend is not seen in the current results from the computer code; the distance in the air increases only 20%. These problems show that this code still has problems, resulting in the current results presented below not including this portion of the code.

IV. Results

At this point there are only preliminary results, with more to come with future work. Even in it's early stages, however, with the test case of the Northrop F-5, the results are promising. Shown below is the figure corresponding to the *RoC* performance metric. It is evident that the matching between the predicted *RoC* and the actual *RoC* is fairly accurate. The errors associated with each data point are shown in the graph to the right in feet per minute (fpm). Note the scaling of both axes.

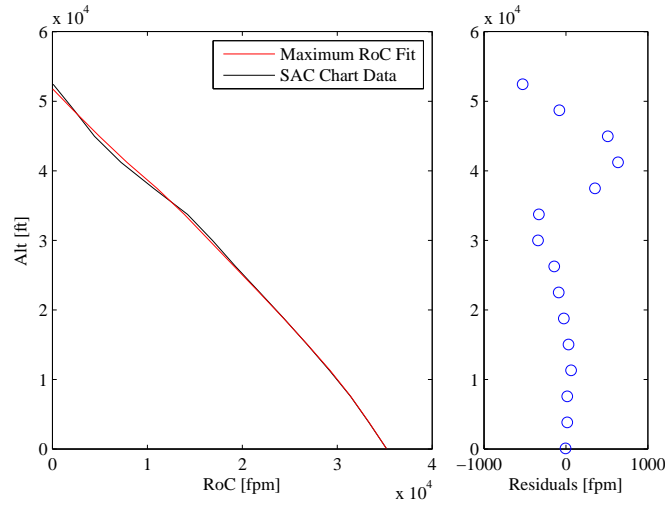


Figure 8. Rate of Climb Results – Northrop F-5

The errors seen in Fig. 8 are all quite small, with the maximum error being about 3.5%. This initial test was also run with the maximum velocity comparison. Maximum error for the maximum velocity worked out to be approximately 4.5%, indicating another great fit.

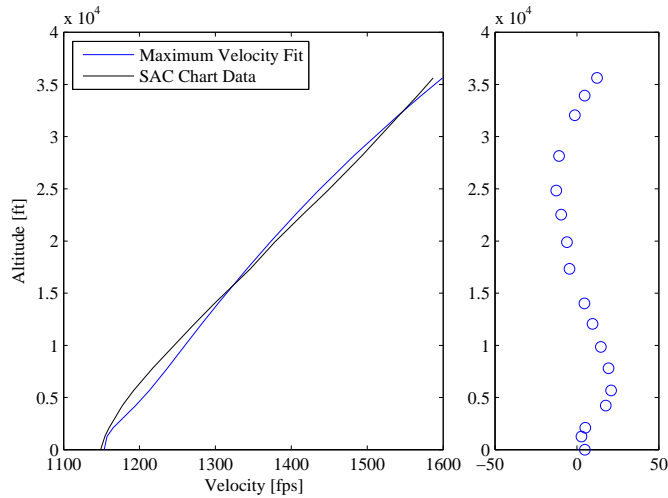


Figure 9. Maximum Velocity Results – Northrop F-5

In terms of engine performance, a comparison was made between the deck found by this program and the F-5 deck used in validation⁹. While this deck is only an approximation, we can see how well this program works, even at these preliminary stages, in Fig. 10. This figure, while not as well correlated as Fig. 5, actually fits the *RoC* curve above much better.

It is also interesting to look at the drag prediction. As was the case with the engine lapse comparison, the drag comparison shows promise, but it by no means perfect. Values of C_L are, from bottom to top, 0.1, 0.3 and 0.5.

There are many possible explanations for this. The biggest factor that stands out to the authors is the narrow range of data included in this study. The maximum *RoC* occurs only between Mach 0.87 and 0.91, depending on altitude, and the Mach range for the maximum velocity is purely supersonic. This eliminates wide ranges of higher C_L values from being used in comparison along with and values of Mach below 0.87. Inclusion of the takeoff routine should

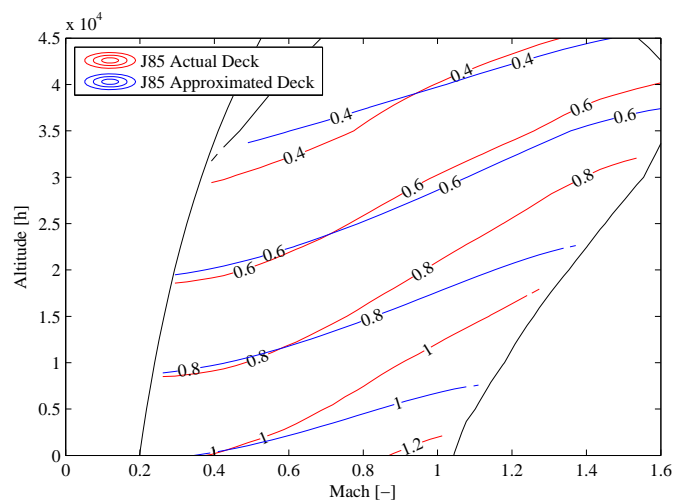


Figure 10. Engine Lapse Comparison – Northrop F-5

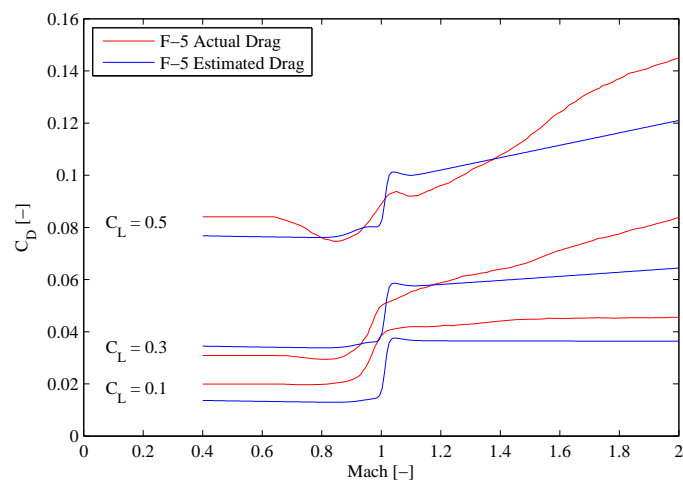


Figure 11. Drag Comparison – Northrop F-5

help this problem, as it will provide information for low Mach numbers and much higher values of C_L ; however, this will only be for sealevel, and a large chunk of data will be still be missing. After the takeoff code is implemented, a radius of turn analysis will be conducted, which will include many additional values spread over multiple Mach and altitude combinations.

V. Conclusion

This program, when fully implemented, will be a useful tool for academia and industry. It can be used in the design process to gather historical data or used by teachers to validate student's results. The wide range of possible applications inside and out of the classroom demonstrate a need for a tool such as this.

The work here is still ongoing, with only preliminary results at this point. Many future developments are planned with this program, including:

- Further code validation
- Increasing code efficiency
- Inclusion of additional performance metrics
- More research into inverse techniques
- Incorporation of error metrics
- Incorporation of propeller models
- More research into drag polars and engine decks

With this future work implemented, a generalized program will be available to solve the inverse problem of aircraft performance. As it stands now the program shows great promise, even with the limited scope of the project thus far.

References

- ¹Company, C. A., "Citation Performance," 2009.
- ²Administration, F. A., "FAA Regulations," 2009.
- ³Heinz W. Engl, C. G., editor, *Inverse and Ill-Posed Problems*, Vol. 4 of *Notes and Reports in Mathematics in Science and Engineering*, Academic Press, Inc., 1987.
- ⁴Richard C. Aster, Brian Borchers, C. H. T., *Parameter Estimation and Inverse Problems*, Elsevier Academic Press, 2005.
- ⁵G.A.F. Seber, C. W., *Nonlinear Regression*, John Wiley & Sons, Inc., 1989.
- ⁶Marquardt, D. W., "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal of the Society for Industrial and Applied Mathematics*, Vol. 11, No. 2, June 1963, pp. 431–441.
- ⁷Dr. Chuan-Tau Edward Lan, D. J. R., *Airplane Aerodynamics and Performance*, Design, Analysis and Research Corporation, 2003.
- ⁸Vinh, N., *Optimal Trajectories in Atmospheric Flight*, Elsevier Scientific Publishing Company, 1981.
- ⁹Vance, C. H., "Standard Aircraft Characteristics Performance of the Northrop F-5E Air Superiority Fighter with Two J85-GE-21 Engines," Tech. rep., Northrop Corporation, December 1976.
- ¹⁰Jack D. Mattingly, William H. Heiser, D. T. P., *Aircraft Engine Design*, AIAA Education, American Institute of Aeronautics and Astronautics, Inc., 2nd ed., 2002.