PEAK OF FLIGHT NEWSLETTER

Issue 651 / May 6th, 2025

Apogee Components, Inc. / ApogeeRockets.com / Colorado Springs, CO

Finding the Coefficient of Drag of Your Rocket

COMING SOON...

PEAK OF FLIGHT NEWSLETTER

Issue 651/ May 6th, 2025

COVER PHOTO

Cluster of Apogee Model Rockets:

Image of our upcoming model rocket kit, the Apogee Mako

FEATURED ARTICLES

The Coefficient of Drag of your ROcket

by Tim Van Milligan

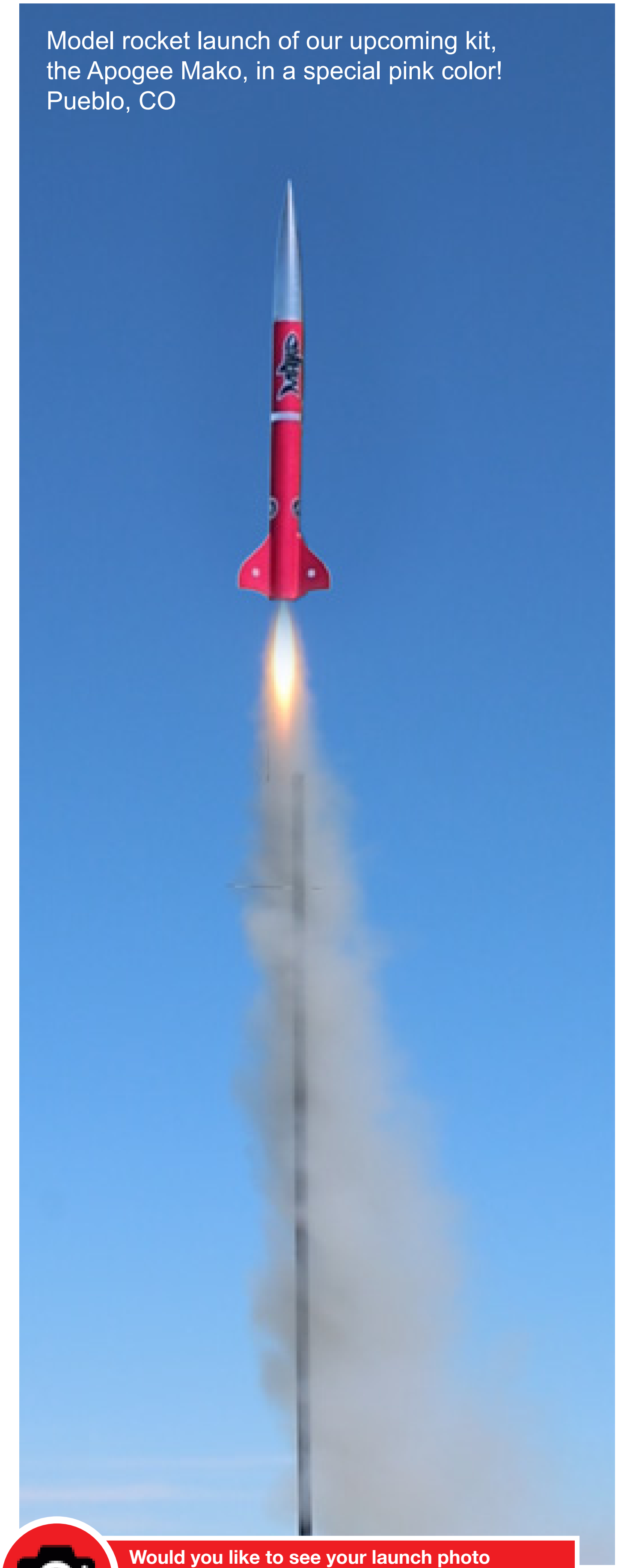
Learn how to determine the Coefficient of Drag of your Model Rocket, and how it can affect your flights.

Apogee Components, Inc.
4960 Northpark Dr.
Colorado Springs, CO 80918
1-719-535-9335
www.ApogeeRockets.com

About this Newsletter

You can subscribe to recieve this e-zine FREE at the Apogee Components website: www.ApogeeComponents.com, or by clicking the link here **Newsletter Sign-Up**

Editor-in-Chief: Tim Van Milligan
Managing Editor: Michelle Mason
Content Editor: Martin Jay McKee
Layout Design: Ogden Sikel



Model rocket launch of our upcoming kit, the Apogee Mako, in a special pink color! Pueblo, CO

Would you like to see your launch photo featured in the *Peak-of-Flight* newsletter? Submit your photo at apogeerockets.com.

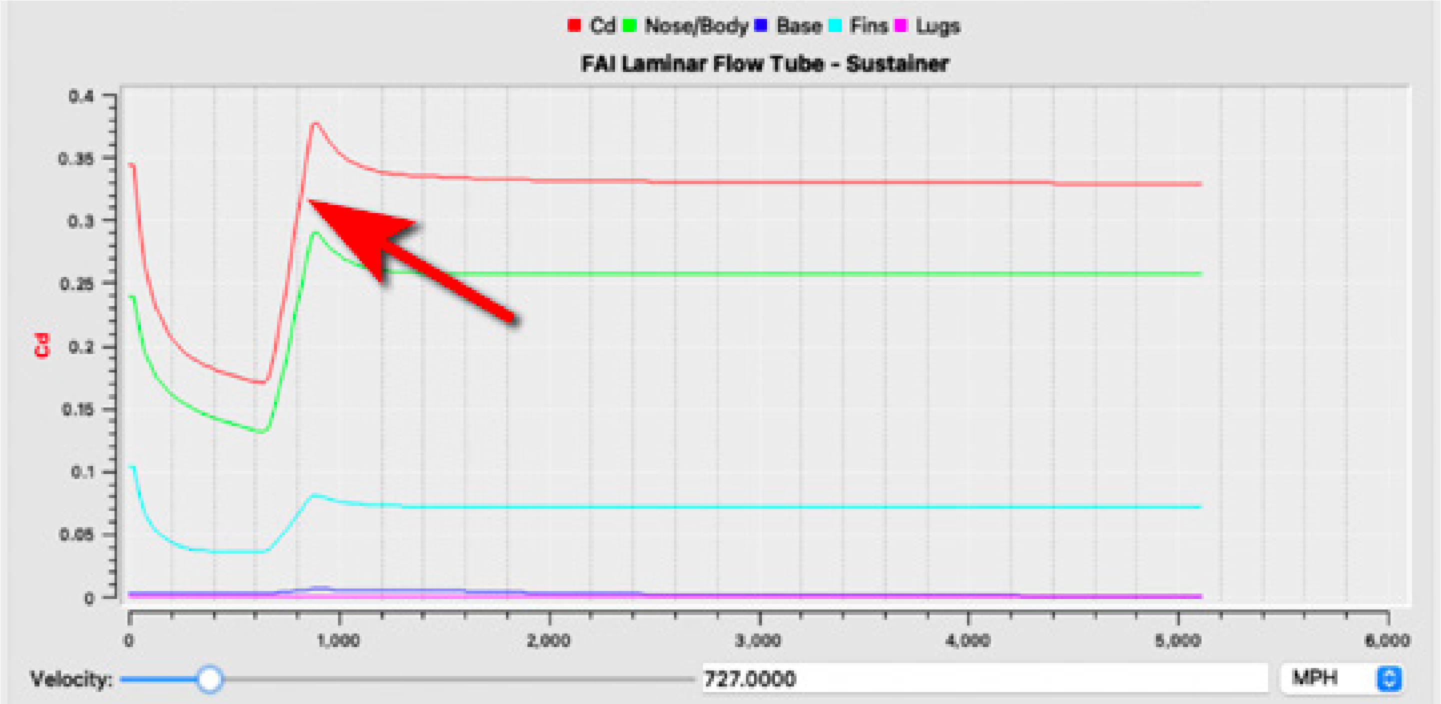
Figure 1: I wanted to know the actual C_d of this streamlined competition rocket

The Coefficient of Drag of Your Rocket

by Tim Van Milligan

The Coefficient of Drag (C_d) is that magic number that defines the efficiency of your rocket. A rocket with a low value C_d is called "more aerodynamic" and will much fly higher and faster than a rocket that has a higher C_d.

This article will give you a step-by-step procedure to accurately measure the C_d of your rocket using data from an altimeter that also records acceleration data. Once you have it, you'll be able to plug this into RockSim and any of your future simulations of the design will be more accurate.



Velocity: 727.0000 MPH

Figure 2: The C_d of a rocket is not actually a constant number as can be seen in this plot of C_d versus speed.

Why is the C_d so Important?

When you are running your RockSim simulations, the software uses the C_d to calculate how high and fast the rocket will fly. But the question that often comes up is, “how accurate is the C_d value that RockSim predicts?” If the value predicted by RockSim is too high, the simulation will come up short compared to an actual flight. And if it is too low, the rocket will sim higher than it flies in real life. Until you put an altimeter into the rocket and fly it, you are unsure of its accuracy, and we want our simulations to be as accurate as possible.

Unfortunately, the C_d is the one number that is actually very hard to predict in advance. In real life, it is not a static number. It varies based on the speed and orientation of the rocket. As the rocket approaches and breaks through the speed of Mach 1, the C_d value rises dramatically, as shown in **Figure 2**. Additionally, if the rocket wobbles (has an angle of attack), the value also increases in sync with the wobble.

Even though it varies, it can still be a useful number for computing your model rocket’s performance. The reason is that for “most” model rockets, we can assume it is a static number. This is true when two caveats hold: **1)** that the rocket doesn’t approach compressible flow (getting close to the speed of sound), and **2)** it is stable and doesn’t wobble too much. And even those rockets that do reach supersonic speeds spend most of their time flying at speeds below Mach 1. So while the predicted altitude and speed will be off, it still allows us to compare the efficiencies of two different rockets.

How C_d was Measured in the Past

The way that C_d was measured in the past was with a method called "back-tracking." Essentially, you'd fly the rocket with an altimeter and record the peak altitude the rocket reached. Then you'd go back into RockSim and input different values for the C_d until the predicted altitude matched the recorded altitude of the actual rocket flight.

Though this does work, what happens if your model rocket weathercocks into the wind? In that case, the rocket doesn't go as high as it could, and the C_d value it comes up with will be overestimated. It will look like the rocket is less efficient compared to flying it on a day where there is no wind, and where the rocket doesn't weathercock.

The other issue is the variability of the rocket motor. We assume that the motor's thrust curve and total impulse will match the certified values for the motor. But those of you that have flown in the American Rocketry Challenge (A.R.C.) event know that motors are rarely reliable to the nominal values published. At least, they are never close enough. So how can you back out a C_d value when the peak altitude will vary from one flight to the next with the same motor?

Inside of our Peak-of-Flight Newsletter issue #303 **(https://www.apogeerockets.com/education/downloads/Newsletter303.pdf)**, I described a process of measuring the C_d by using a recording altimeter that also records the velocity of the rocket using an accelerometer. That article was overly complex, and if we go back to first principles of physics, we can come up with a way to directly calculate the C_d of the rocket—that is what I'll describe here.

How Recording Altimeters Give Us Better Data

Recording altimeters are constantly measuring the altitude and speed of the rocket and storing the data in a spreadsheet that can be downloaded and transferred to your computer. The advantage of continuous recording of the data is that we can compute the C_d hundreds of times for each flight, whereas back-tracking with the old method gave just one C_d value. You can then average them, and your confidence in pinning the value down goes way up. One single flight is better than dozens of flights using the old method.

When doing these calculations, what we want is the speed of the rocket, and more specifically, the speed AFTER motor burnout. In this way, we eliminate the variability of the rocket motors thrust curve and Total Impulse. That gives us a more pure number that we can count on for future simulations.

What is the Formula for the Cd?

We're going to get the C_d from the Drag formula:

Drag (D) = ½ Rho C_d A V^2
These terms are as follows:
Rho = density of air
S = the reference area of the rocket (which for model rockets is the area at the base of the nose cone. This is the area of a circle, which we can easily compute.)
V = Velocity of the rocket.

Drag is a "force," and according to Newton's 2nd law of motion, the formula for a force is:
F = m a
Where:
a = acceleration
m = mass of the rocket

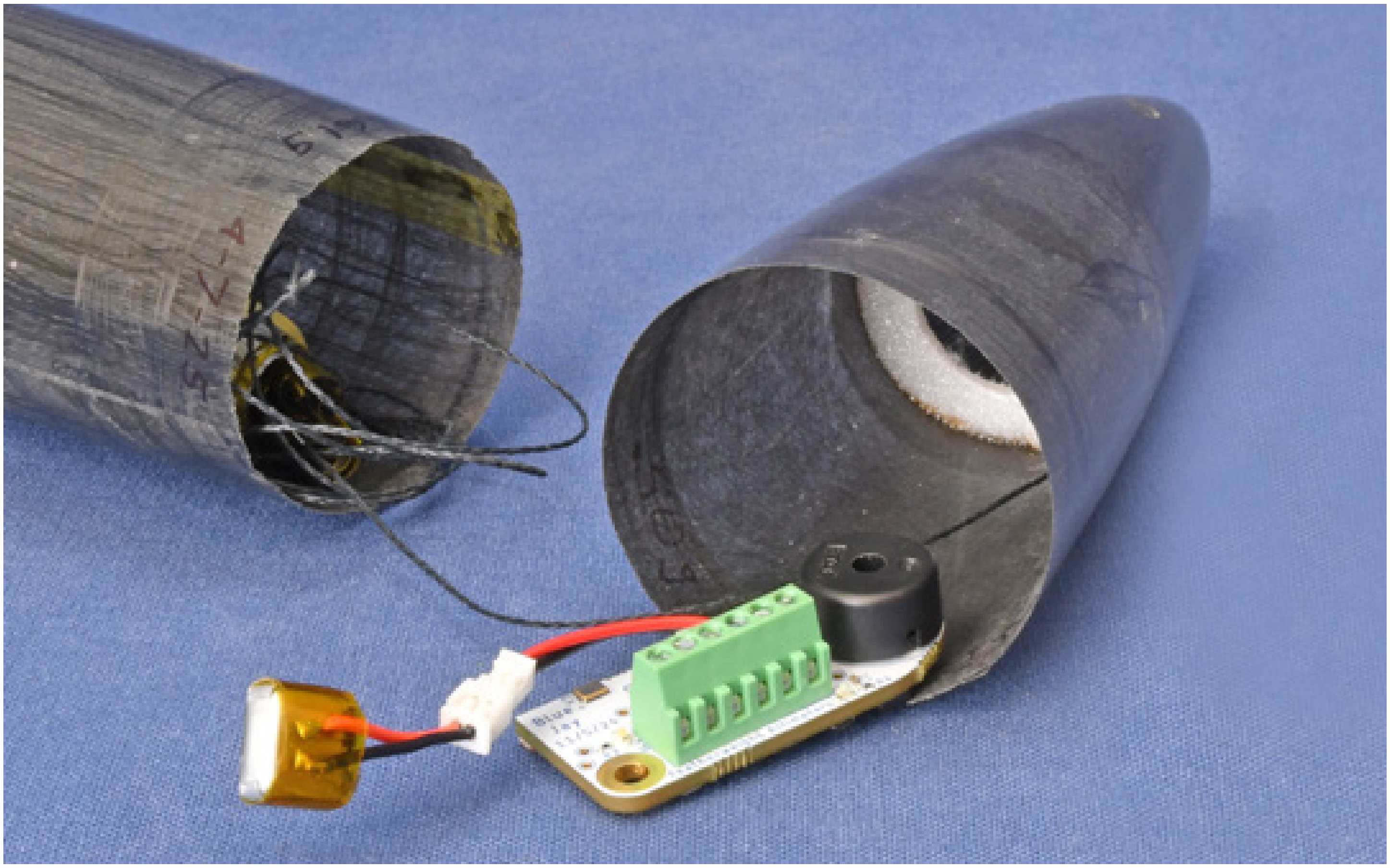
What we'll do is substitute the **F** for **D**, so the new equation is:
m a = ½ Rho C_d S V2

Then we'll rearrange the equation to isolate the **C_d**
C_d = (2 m a) / Rho S V2

We have to remember, the acceleration is affected by two forces acting on the rocket: Drag and the force of gravity are both slowing the rocket down. But our accelerometer on our rocket is measuring total acceleration, which is the sum of both these forces. We need to subtract out the force from gravity. The new equation for **C_d** is:
C_d = (2 m {-a - g}) / Rho S V2

As shown above, "**a**" in this case is the deceleration force due to drag, and g is the force of gravity (9.81 m/s2). By using a negative "**a**", the bracketed value is positive because **$-a - g > 0$** during upward coast.

We now have a formula for calculating the C_d of the rocket. All we need is the rocket velocity and the acceleration at that velocity.



Peeking Into the Altimeter Spreadsheet

At this point, we need to peek into the spreadsheet data that was recorded by the altimeter. In my example that I'm going through here, the altimeter used the **Featherweight Blue Jay Plus**.This is a very small device that has both a pressure sensor to measure the altitude and an IMU that measures the acceleration of the rocket in three directions (vertical, cross range, down range). It is also relatively cheap, which makes it great for this type of experiment.

When you download the data from the altimeter to your computer and open the .csv file, you'll see a massive amount of information. It's almost overwhelming. Note: here is a sample altimeter data file that you can play around with: **https://www.apogeerockets.com/downloads/PDFs/Cd-Test-LR_04-2025_13.csv**. But don't worry, we don't need it all—we're going to purge most of it.

But first, we need to get the lay of the land and explore what's in it. The column headers at the top give us some idea, but they don't always tell us the units. Everything is recorded in the "English" units. So the units for velocity are feet/second.

Figure 3 - The Blue Jay Plus altimeter fits easily inside the nose cone tip. A cotton ball was stuffed in behind to cushion it, and a piece of tape over the foam centering ring held it in place during the flight.

That is important because we'll have to do a conversion to metric to run our equation. I suppose you can use English units, but then you have other conversions to do. To convert ft/sec to m/sec, we'll multiply by .3048.

The columns that are important to us for this project are labeled: Flight Time and Velocity_Up. That's it, just two columns from this huge collection of data recorded!

We're going to assume that our rocket is going UP and not weathercocking into the wind, which would give us velocity downrange or crossrange. If your rocket isn't flying straight up… you've got other problems - so fix them first. We can tolerate a little bit of non-vertical, but not rockets going off at an angle, or our C_d value will be way off using the simple equation here.

Computing the Acceleration

Notice that the altimeter isn't storing the acceleration values. That was a choice made by Featherweight. Instead of acceleration, what modelers often want to see is the velocity (speed) of the rocket. So the device is actually computing the velocity onboard the rocket itself, from the acceleration data, and just storing the velocities in the spreadsheet.

What this means to us is that we have to work backwards and derive the acceleration values from the velocity data. To do this, we'll use three velocity points. The formula is:

$$a_i = (v_{i+1} – v_{i–1}) / (2 \Delta t)$$

What this means, is that the acceleration at time **"i"** is computed by finding the velocity at the next data point after time **i** (which is the **{i + 1}**) and subtracting the velocity at the previous data point **{i - 1}**. This is divided by 2 multiplied by the time increment. It is a little confusing, but it works. Let's go through a simple example:

Time	Velocity
0.64	85.6488 m/s
0.66	84.4296 m/s
0.68	83.5152 m/s

Here the time increment between data points is 0.02 seconds. The acceleration at time #2 (which is at 0.66 seconds), is found using this equation:

a = (83.5152 m/s - 85.6488 m/s) / 2 * .02 sec

a = -53.34 m/s2

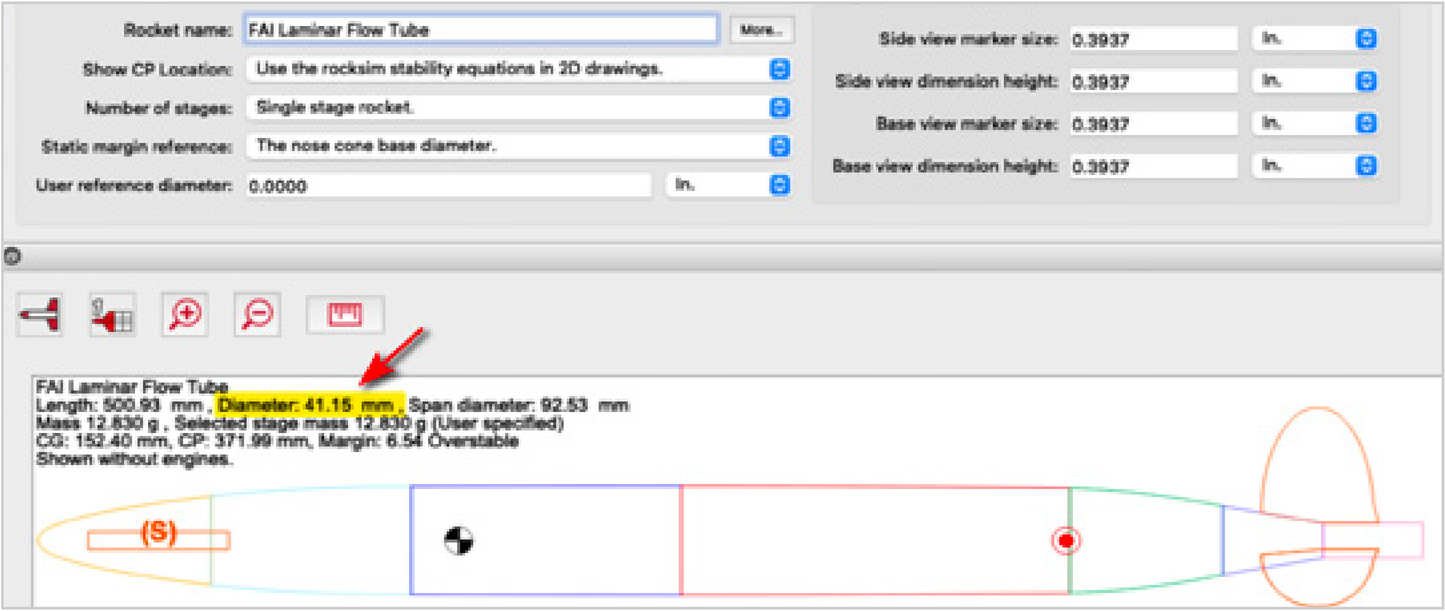
At this point, we have all the information we need to find the Coefficient of Drag. But to make the process easier, I'll create a spreadsheet to do all the computations for us. This spreadsheet is found at: **https://docs.google.com/spreadsheets/d/1zH2GC2jzvzxt480Z9gffzZcNBag8htwj7gHqQL7rBok/edit?usp=sharing.**

We'll also need three other pieces of information. We're going to need the diameter of the rocket, the density of the air the rocket is flying through, and the burnout mass of the rocket (after all the propellant has been consumed).

Let's get that information first, by opening up RockSim.

Figure 4 shows the location in RockSim where you can find the diameter of your rocket that we'll need. I suggest going into the preferences and changing the units of length to mm so you can just easily record the diameter.

Figure 4: You can find the diameter in the 2D image of your model rocket.



Rocket name: FAI Laminar Flow Tube

Show CP Location: Use the rocksim stability equations in 2D drawings.

Number of stages: Single stage rocket.

Static margin reference: The nose cone base diameter.

User reference diameter: 0.0000 In.

Side view marker size: 0.3937 In.

Side view dimension height: 0.3937 In.

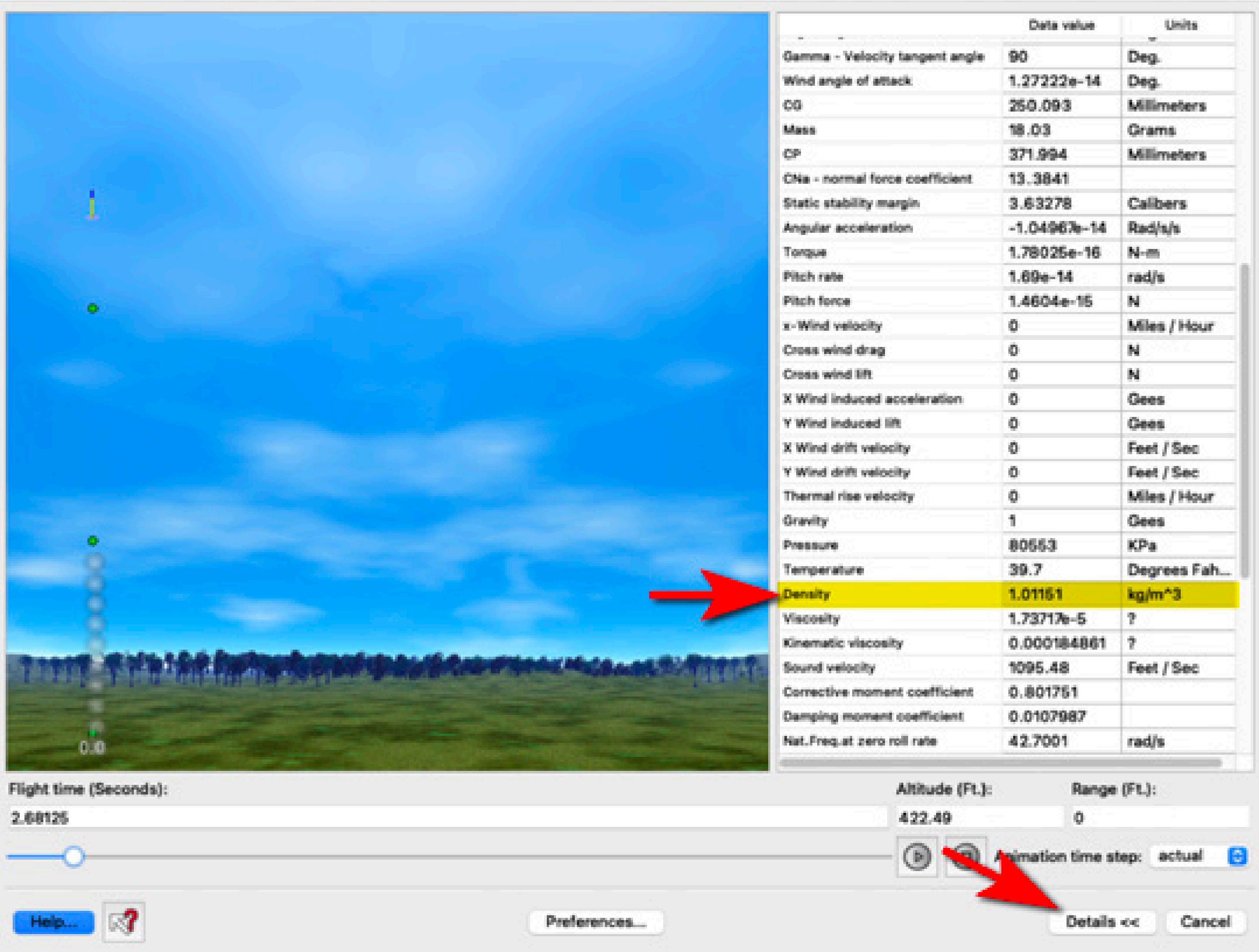
Base view marker size: 0.3937 In.

Base view dimension height: 0.3937 In.

FAI Laminar Flow Tube
Length: 500.93 mm , Diameter: 41.15 mm , Span diameter: 92.53 mm
Mass 12.830 g , Selected stage mass 12.830 g (User specified)
CG: 152.40 mm, CP: 371.99 mm, Margin: 6.54 Overstable
Shown without engines.

PEAK OF FLIGHT NEWSLETTER
Issue 651 / May 6th, 2025
www.ApogeeRockets.com

Find the Coefficient of Drag of your Rocket



	Data value	Units
Gamma - Velocity tangent angle	90	Deg.
Wind angle of attack	1.27222e-14	Deg.
CG	250.093	Millimeters
Mass	18.03	Grams
CP	371.994	Millimeters
CNa - normal force coefficient	13.3841	
Static stability margin	3.63278	Calibers
Angular acceleration	-1.04967e-14	Rad/s/s
Torque	1.78025e-16	N-m
Pitch rate	1.69e-14	rad/s
Pitch force	1.4604e-15	N
x-Wind velocity	0	Miles / Hour
Cross wind drag	0	N
Cross wind lift	0	N
X Wind induced acceleration	0	Gees
Y Wind induced lift	0	Gees
X Wind drift velocity	0	Feet / Sec
Y Wind drift velocity	0	Feet / Sec
Thermal rise velocity	0	Miles / Hour
Gravity	1	Gees
Pressure	80553	KPa
Temperature	39.7	Degrees Fah...
Density	1.01161	kg/m^3
Viscosity	1.73717e-5	?
Kinematic viscosity	0.000184861	?
Sound velocity	1095.48	Feet / Sec
Corrective moment coefficient	0.801751	
Damping moment coefficient	0.0107987	
Nat.Freq.at zero roll rate	42.7001	rad/s

Flight time (Seconds): 2.68125

Altitude (Ft.): 422.49

Range (Ft.): 0

Animation time step: actual

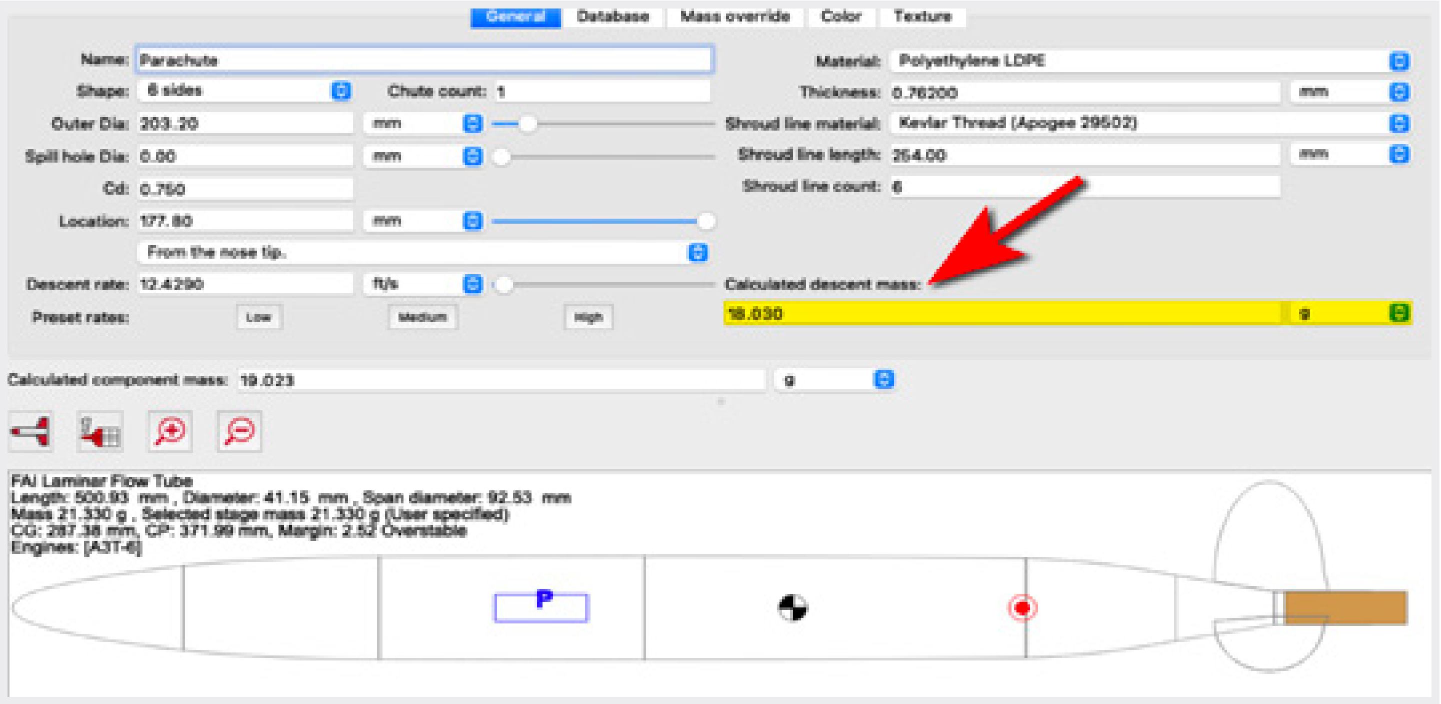
Figure 5: You can pick off the density of the air from the details of the 2D flight profile in RockSim.

The air density, rho, is the hardest parameter to find. The density of air depends on the altitude above sea level, the temperature of the air, and the air pressure on the day of the launch. Fortunately, RockSim does all the equations to determine the air density. But you have to run a simulation to find it.

What I'd suggest is you first run your simulation, and display a 2D flight launch simulation. On the bottom right, there is a button called "Details." Click on that, and it will display all the details of the simulation in a table on the right hand side of the screen, such as shown in **figure 5.**

Since the air density is constantly changing while the rocket ascends in the sky, you need to decide what altitude to pick the value. My suggestion is to move the time scrubber on the bottom of the screen until you're sure the motor has burned out. You don't have to get too exact, because the density isn't making dramatic changes unless your rocket is going to orbit. You can get carried away, but I wouldn't worry about it. As we mentioned at the beginning, the C_d value varies anyway, so we'll never be exact.

We'll also need the burnout weight of the rocket. For this, it is best to actually measure it with a scale after you launch the rocket.



Name: Parachute

Shape: 6 sides — Chute count: 1

Outer Dia: 203.20 mm

Spill hole Dia: 0.00 mm

Cd: 0.750

Location: 177.80 mm

From the nose tip.

Descent rate: 12.4290 ft/s

Preset rates:

Material: Polyethylene LDPE

Thickness: 0.76200 mm

Shroud line material: Kevlar Thread (Apogee 29502)

Shroud line length: 254.00 mm

Shroud line count: 6

Calculated descent mass: 18.030 g

Calculated component mass: 19.023 g

FAI Laminar Flow Tube
Length: 500.93 mm , Diameter: 41.15 mm , Span diameter: 92.53 mm
Mass 21.330 g , Selected stage mass 21.330 g (User specified)
CG: 287.38 mm, CP: 371.99 mm, Margin: 2.52 Overstable
Engines: [A3T-6]

Figure 6 - The parachute design screen has a field that will show you the descent mass of the rocket.

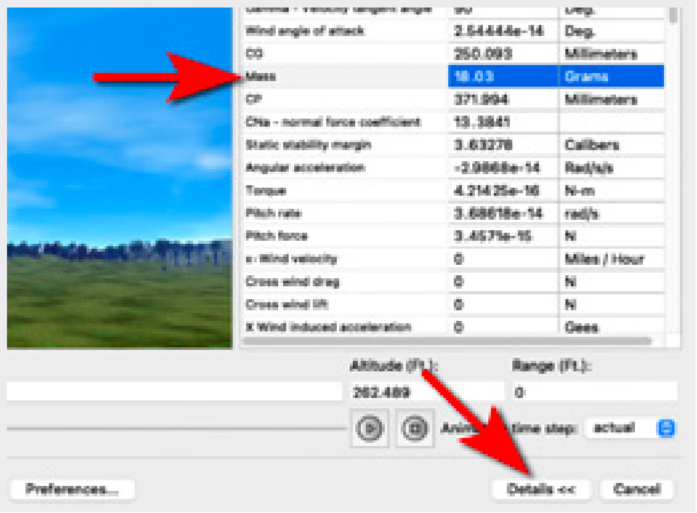
If you forgot to do that, the next best thing is to get the information from RockSim. If your RockSim design has a parachute in it, the design screen for the parachute will list the descent mass of the rocket. It knows that the descent mass is the total mass, minus the mass of the propellant. All that information is within RockSim, so it knows what the descent mass should be. You'll find the information in two places. First is on the parachute design screen as shown in **Figure 6.**

If your rocket has a streamer instead of a parachute, you'll have to "show details" on the 2D flight profile of a launch, and then scrub through the trajectory towards the end and look at the mass of the rocket **(Figure 7).** Just before it touches down will be an accurate value for the descent mass.

With these numbers, the burnout mass, the reference diameter and the air density, we can now start filling out the spreadsheet to calculate the C_d. Open up the spreadsheet, and drop those numbers into the yellow data fields at the top of the table.

Next, open up the csv file with the data recorded by your data logging altimeter. While there is a lot of data stored in this file, we'll only be concerned with two columns, the "Flight_Time," and the "Velocity_Up".

I personally resave the file with a different name, so I can edit the file to make it more manageable. If something goes wrong with the file you're editing, you still have the other file as a backup.

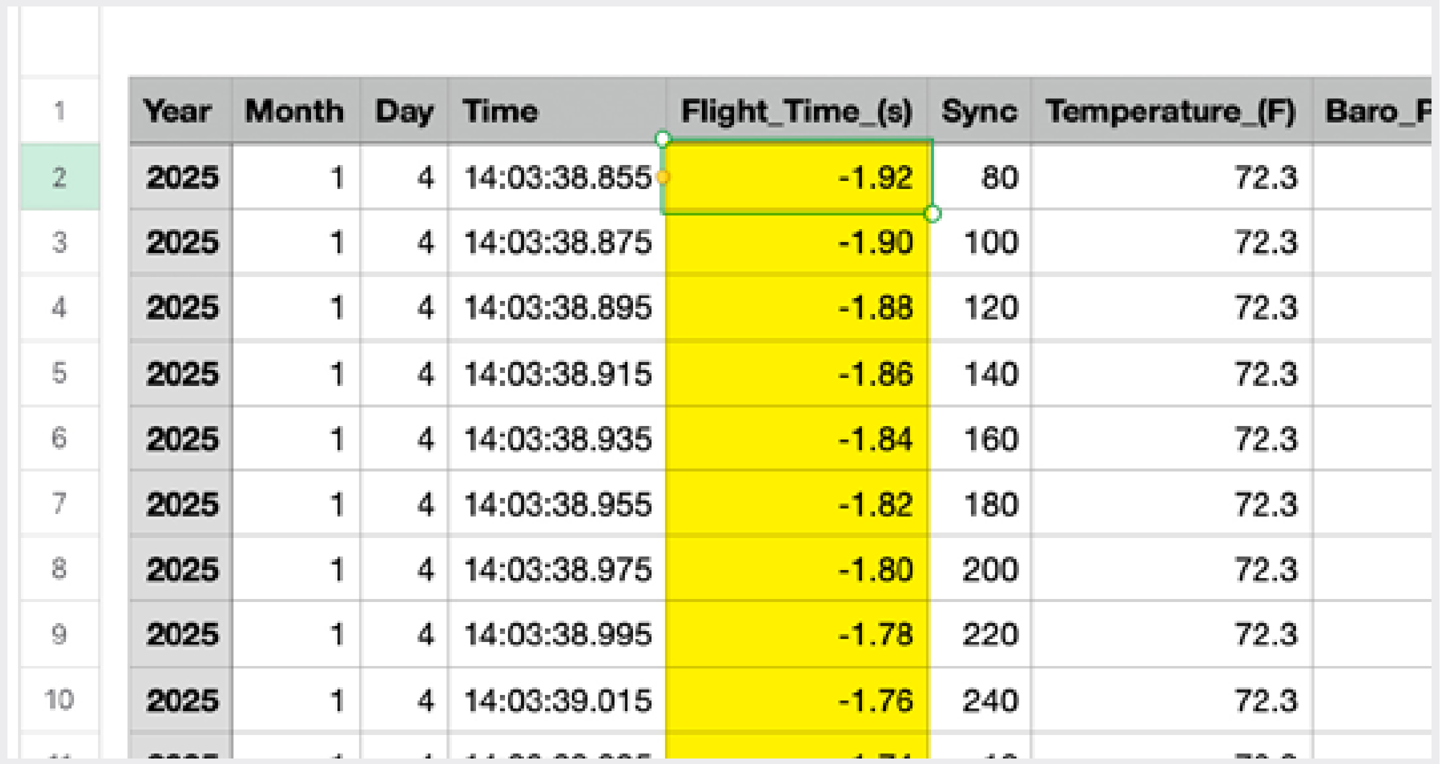


Gamma - Velocity tangent angle	90	Deg.
Wind angle of attack	2.54444e-14	Deg.
CG	250.093	Millimeters
Mass	18.03	Grams
CP	371.994	Millimeters
CNa - normal force coefficient	13.3841	
Static stability margin	3.63278	Calibers
Angular acceleration	-2.9868e-14	Rad/s/s
Torque	4.21425e-16	N-m
Pitch rate	3.68618e-14	rad/s
Pitch force	3.4571e-15	N
x- Wind velocity	0	Miles / Hour
Cross wind drag	0	N
Cross wind lift	0	N
X Wind induced acceleration	0	Gees

Altitude (Ft.): 262.489

Range (Ft.): 0

Figure 7 - You can also get the descent mass from the Details in the 2D flight profile screen. This is good for streamers.

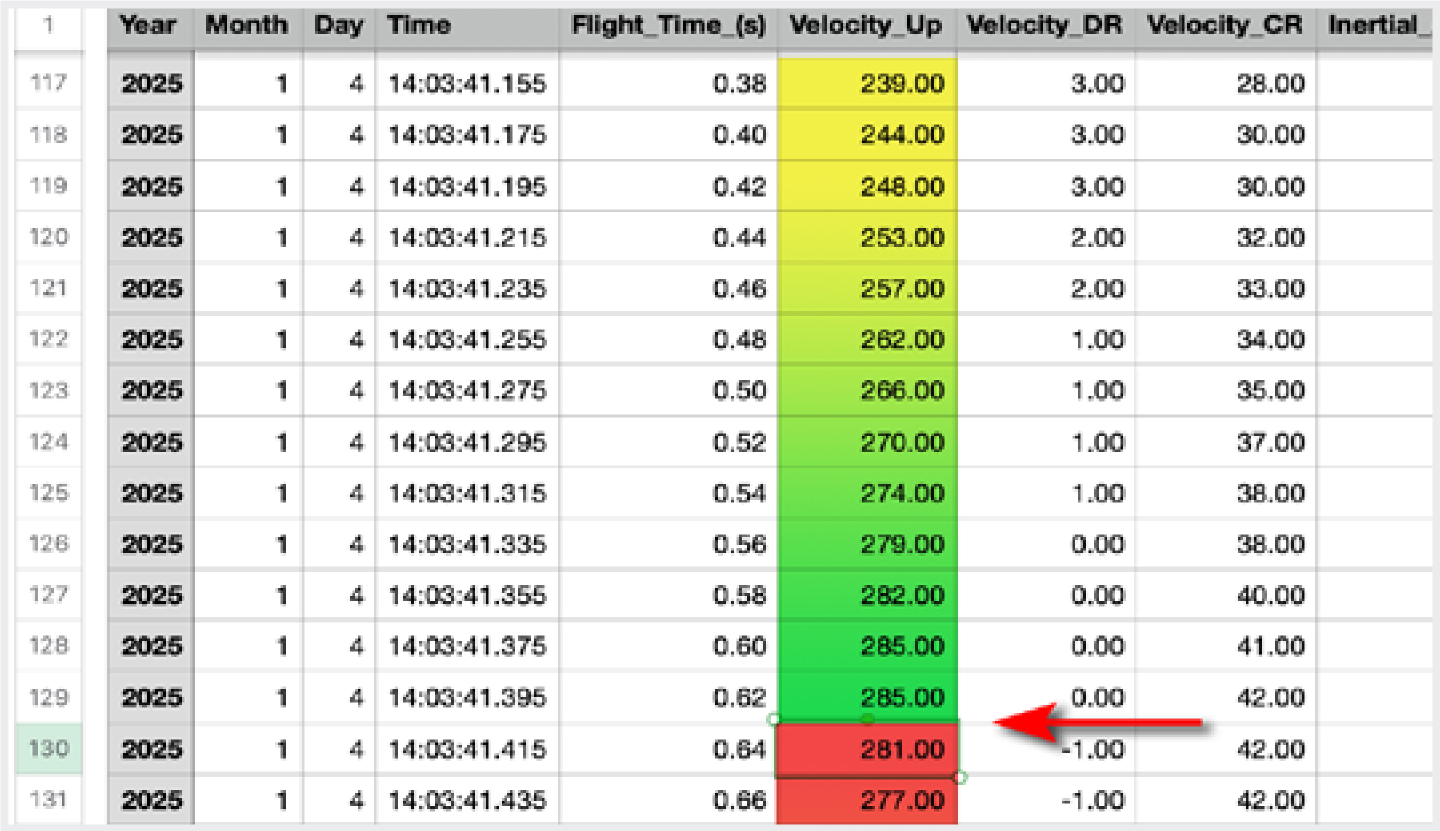


1	Year	Month	Day	Time	Flight_Time_(s)	Sync	Temperature_(F)	Baro_F
2	2025	1	4	14:03:38.855	-1.92	80	72.3	
3	2025	1	4	14:03:38.875	-1.90	100	72.3	
4	2025	1	4	14:03:38.895	-1.88	120	72.3	
5	2025	1	4	14:03:38.915	-1.86	140	72.3	
6	2025	1	4	14:03:38.935	-1.84	160	72.3	
7	2025	1	4	14:03:38.955	-1.82	180	72.3	
8	2025	1	4	14:03:38.975	-1.80	200	72.3	
9	2025	1	4	14:03:38.995	-1.78	220	72.3	
10	2025	1	4	14:03:39.015	-1.76	240	72.3	

Figure 8 - Delete all the rows from the table where the time is negative. This is when the rocket is sitting on the launch pad.

Notice that the Flight_Time column doesn't start at time=0 (See **Figure 8**). It starts at a negative time before liftoff. Featherweight devices do that so that it doesn't accidentally lose any data right at ignition when things start to change dramatically. So what I do is to delete the rows where the Flight_Time has negative data. As long as the rocket is sitting on the pad, we're not going to get any meaningful data.

Next, we want to look down through the "Velocity_Up" column. What we're looking for is the peak velocity that the rocket achieves. As long as the rocket motor is burning, the velocity of the rocket will continue to increase. So when the velocity peaks, we are really close to the burn-out of the rocket motor **(Figure 9)**.



1	Year	Month	Day	Time	Flight_Time_(s)	Velocity_Up	Velocity_DR	Velocity_CR	Inertial_
117	2025	1	4	14:03:41.155	0.38	239.00	3.00	28.00	
118	2025	1	4	14:03:41.175	0.40	244.00	3.00	30.00	
119	2025	1	4	14:03:41.195	0.42	248.00	3.00	30.00	
120	2025	1	4	14:03:41.215	0.44	253.00	2.00	32.00	
121	2025	1	4	14:03:41.235	0.46	257.00	2.00	33.00	
122	2025	1	4	14:03:41.255	0.48	262.00	1.00	34.00	
123	2025	1	4	14:03:41.275	0.50	266.00	1.00	35.00	
124	2025	1	4	14:03:41.295	0.52	270.00	1.00	37.00	
125	2025	1	4	14:03:41.315	0.54	274.00	1.00	38.00	
126	2025	1	4	14:03:41.335	0.56	279.00	0.00	38.00	
127	2025	1	4	14:03:41.355	0.58	282.00	0.00	40.00	
128	2025	1	4	14:03:41.375	0.60	285.00	0.00	41.00	
129	2025	1	4	14:03:41.395	0.62	285.00	0.00	42.00	
130	2025	1	4	14:03:41.415	0.64	281.00	-1.00	42.00	
131	2025	1	4	14:03:41.435	0.66	277.00	-1.00	42.00	

Figure 9 - Find where the peak speed of the rocket occurs. Delete all the data prior to the peak speed, as this is where the rocket motor is burning. We only want coast phase data.

We don't want to use data during the thrust phase, because instead of two forces acting on the rocket (gravity and drag), we'll also have a thrust force. That complicates our calculations for C_d, so we only want data that is past motor burnout. The rocket needs to be in the coast phase.

In actuality, peak thrust typically occurs just prior to burnout because for most motors the thrust tails off gradually. You can see this in RockSim if you run a simulation and then plot out a graph of thrust and velocity versus time, as shown in **Figure 10.** The peak velocity is almost always before motor burnout by a little bit. It does depend on the motor, so if you have a rocket motor that has a long burn, you'd better check where burnout actually occurs by looking at RockSim. Because if you are calculating the C_d while the motor is producing thrust, it is going to skew the results. It is better to play it safe and go well past the peak speed in the .csv file.

To reiterate, delete all the rows from the .csv file that are prior to peak speed. You don't want them in the calculations.

After that, you can delete all the data from the spreadsheet that occurs after the rocket reaches apogee (see Figure 11).

Look at the data in the Velocity_Up column, and note where it changes from a positive number to zero and then starts going negative. This is the apogee point of the rocket. Once you note that point, delete all the rows after apogee.

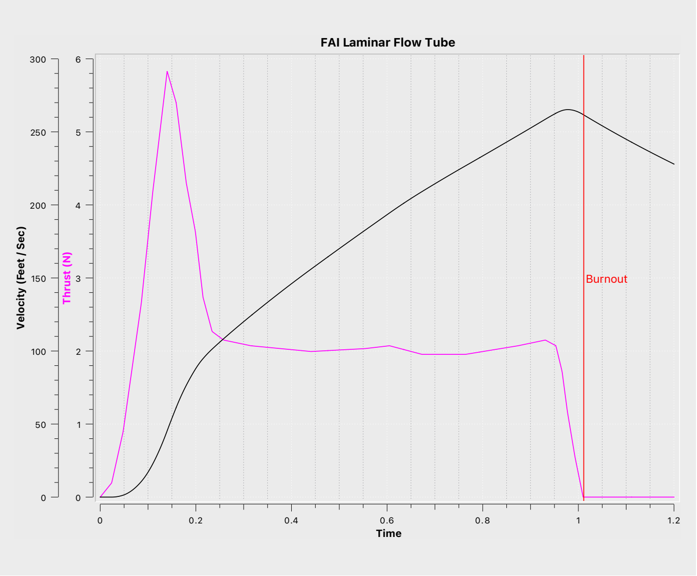


Figure 10 - The peak velocity is usually just prior to motor burnout. Double check with RockSim to make sure you know at what time burnout occurs.

Note, if your recovery device is ejected early, prior to apogee, you'll still see the point in the data where the speed dramatically changes instead of the nice slow-down that my chart shows.

At this point, our data table has been cleaned up and all the excess data has been purged to avoid confusion. Now we'll transfer the useful information over to our other spreadsheet that does the calculations for us.

Select and copy all the remaining data from the Flight_Time column and paste it in the CD Measurement spreadsheet in the yellow Flight_Time (s) column. Then do the same for the Velocity_Up column and paste it into the Vel ft/s column.

If everything works correctly, the equations in the other columns should automatically update and calculate the C_d at each point during the flight.

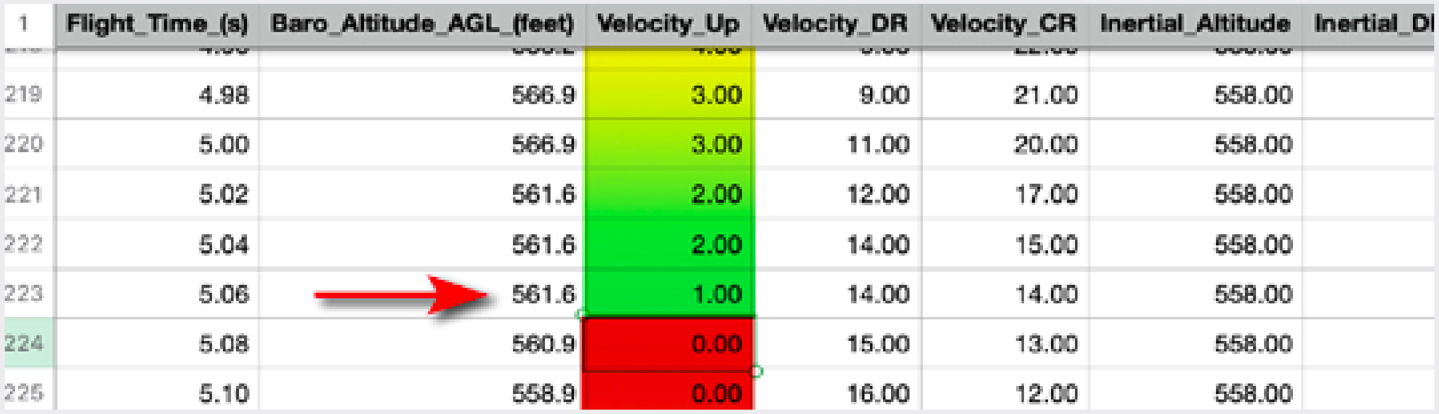
SATURN 1B
- Stands Over Three Feet Tall
- Most Authentic Saturn 1B Made
- 29mm Motors Compatible
SHOP NOW
USA

PEAK OF FLIGHT NEWSLETTER

Find the Coefficient of Drag of your Rocket

Issue 651 / May 6th, 2025

www.ApogeeRockets.com



1	Flight_Time_(s)	Baro_Altitude_AGL_(feet)	Velocity_Up	Velocity_DR	Velocity_CR	Inertial_Altitude	Inertial_DI
219	4.98	566.9	3.00	9.00	21.00	558.00	
220	5.00	566.9	3.00	11.00	20.00	558.00	
221	5.02	561.6	2.00	12.00	17.00	558.00	
222	5.04	561.6	2.00	14.00	15.00	558.00	
223	5.06	561.6	1.00	14.00	14.00	558.00	
224	5.08	560.9	0.00	15.00	13.00	558.00	
225	5.10	558.9	0.00	16.00	12.00	558.00	

Figure 11 - Delete all the columns after the rocket reaches apogee, which is where the Velocity_Up is no longer a positive number.

If you have a lot of data, you may need to copy the formulas on the right to the blank columns to get the C_d values for every moment during the upward coast phase of the rocket.

My final table for the example rocket I did is in **figure 12.**

Interpreting the Data - What does it mean?

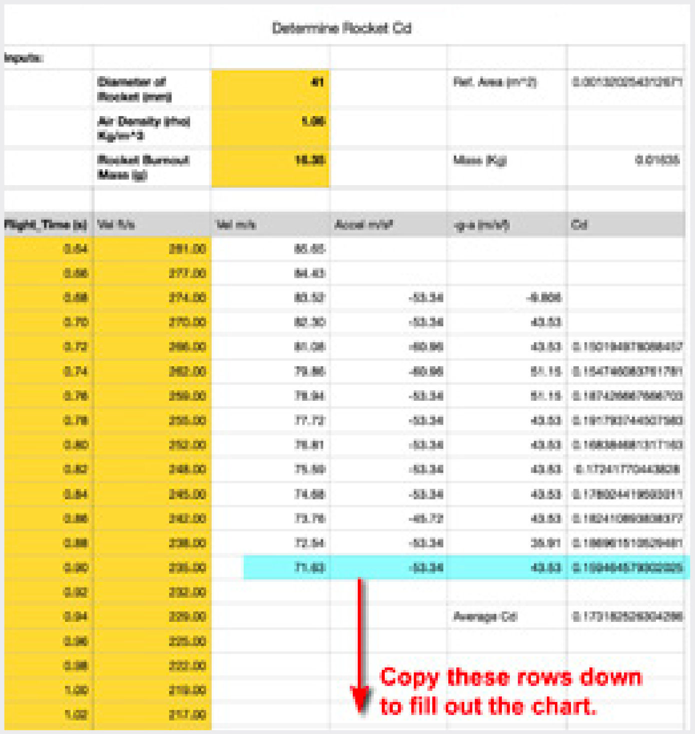
In order to understand all my examples here, note that the data I collected for this article was from a highly optimized competition rocket **(Figure 1)**. I did everything I could to reduce the drag of the rocket. It was glass smooth and the shape was such that I knew it would have a low coefficient of drag. I had no idea that it would be this low (under 0.2). Most model rockets that you will fly will have a much higher C_d. A good reference point is between 0.35 and 0.75. Some very draggy models will have a C_d of >1.0.

Side note: If you want to build a low drag rocket like the one shown here, I'm all for it. I don't hoard my secrets… because I want to see you do even greater things than what I've done. we've got lots of articles in our Apogee newsletter archives about competition low drag model rockets **(https://www.apogeerockets.com/Peak-of-Flight?pof_list=topics&m=education).**

One thing you'll notice within the C_d chart, is that the value fluctuates. It never says steady. There are reasons for this.

First, the accelerometer that measures the velocity isn't perfect. There is sensor noise *(see Peak-of-Flight Newsletter #649 - https://www.apogeerockets.com/education/downloads/Newsletter649.pdf)* that will alway be present. Plus, it seems to lag a little bit when taking readings. Because these speed and velocity numbers are jittery, the C_d is going to fluctuate.

Second, the data we're taking in is from velocity data, not the original acceleration readings that the sensor outputs. The velocity is being computed on the device before being stored in memory, instead of storing accelerometer readings. It is probably an issue of limited available memory on the Blue Jay Plus rather



Determine Rocket Cd

Inputs:					
	Diameter of Rocket (mm)	41		Ref. Area (m^2)	0.001320254312671
	Air Density (rho) Kg/m^3	1.06			
	Rocket Burnout Mass (g)	16.35		Mass (Kg)	0.01635
Flight_Time (s)	Vel ft/s	Vel m/s	Accel m/s²	-g-a (m/s²)	Cd
0.64	281.00	85.65			
0.66	277.00	84.43			
0.68	274.00	83.52	-53.34	-9.806	
0.70	270.00	82.30	-53.34	43.53	
0.72	266.00	81.08	-60.96	43.53	0.150194978098457
0.74	262.00	79.86	-60.96	51.15	0.154749083761781
0.76	259.00	78.94	-53.34	51.15	0.187426667666703
0.78	255.00	77.72	-53.34	43.53	0.191793744507583
0.80	252.00	76.81	-53.34	43.53	0.168384481317163
0.82	248.00	75.59	-53.34	43.53	0.17241770443808
0.84	245.00	74.68	-53.34	43.53	0.178024419593011
0.86	242.00	73.76	-45.72	43.53	0.182410893838377
0.88	238.00	72.54	-53.34	35.91	0.189961510529481
0.90	235.00	71.63	-53.34	43.53	0.159464579302025
0.92	232.00				
0.94	229.00			Average Cd	0.173182526304286
0.96	225.00				
0.98	222.00				
1.00	219.00				
1.02	217.00				

Copy these rows down to fill out the chart.

Figure 12 - Once the Time and Velocity Data is transferred over to the new chart, it should automatically determine the Cd at every point during the upward coast phase.

than a failure of the manufacturer. But what this means is that the velocity is being rounded to the nearest foot/sec, which is reducing the accuracy of the data. We'd rather have the raw acceleration data, and if you do have a different altimeter device that outputs acceleration (such as the Blue Raven), then the spreadsheet can be adjusted to calculate velocity.

Third, the rocket is probably not going perfectly straight up. It is probably wobbling ever so slightly, even if it looks like it was riding on a laser beam into the sky. When it is at a higher angle of attack, the C_d value is going to be higher than when the rocket is flying perfectly straight.

Fourth, we are assuming that the density of the air is the same throughout the entire flight. But it isn't. In fact, it is actually getting lower the higher up the rocket goes. Furthermore, it may be flying through bubbles of warm or colder air that have densities that may be drastically different.

Fifth, as you'll notice from the data, when the velocity of the rocket slows down, the C_d values start to get really erratic. You can see in my example data some negative numbers, which are nonsense. The reason for this is that we're measuring the Drag force on the rocket, and as you know from the fundamental Drag equation, the force goes up by the square of the velocity. When slowing down, the drag force drops exponentially. What this means is that it is MUCH harder to measure drag (though deceleration) at slower speeds. We can't measure it accurately, so we can't compute the C_d accurately at slower speeds.

The key takeaway is this: you get the more accurate C_d numbers at higher speeds, such as just after the rocket motor burns out.

Therefore, averaging the C_d over the entire coast phase is likely pointless because of the low-speed measurement problem. I've put an average in the chart, but it can be ignored. You might average the C_d for a certain period of time when the standard deviation is pretty low. That would probably be your best option.

What Do You Do With the Data?

Now that we've computed the C_d from actual flight data, what do we do with it?

Well, I think this information is worth its weight in gold. You've really done yourself a huge favor by throwing that altimeter into the rocket and backing out the C_d. So I'd like to congratulate you.

The real purpose is that if you have a good C_d number for a rocket, it will make your next simulation in RockSim more accurate. It is the one number that we had to estimate, and now we have a number that is nailed down.

So what you're going to do with it is use the C_d override in RockSim and put this number there.

You'll be more confident that the actual results will more closely match what RockSim predicts. And second, you better determine what ejection delay you should be using for your rockets. You can't really know for certain the optimum ejection delay will be if you don't know exactly how high the rocket is going to fly.

Conclusion

As I was writing this, I thought to myself that this is probably a new feature we could build into RockSim-Pro. In that software, we're already allowing recorded altimeter data to be imported to "replay" the flight. There isn't any reason why we couldn't also use the same data to find the C_d of the rocket. It would be really helpful and reduce the amount of work that users have to do.

I really like this method much better than doing the old method of just backtracking based on a single altitude that the rocket reached. The big advantage is that we don't have to wait for a perfectly windless day to fly the rocket to get a perfectly straight flight. We can fly in breezy conditions and still get useful information.

And as mentioned, you get better data when you fly your rocket at higher speeds. So, use a higher thrust rocket engine when doing this. That also helps to assure that the rocket goes straight, even in breezy conditions.

Absolutely give this process a try! You'll gain valuable insights into your rocket's performance and dramatically improve the accuracy of your RockSim simulations. By measuring the Coefficient of Drag directly from flight data, you move beyond estimations and obtain a real-world value that will make your future predictions much more reliable. Plus, it's a fantastic learning experience that deepens your understanding of aerodynamics and rocket science. Don't be intimidated by the calculations; the spreadsheet tool simplifies the process, and the rewards of having precise data are well worth the effort. You'll be able to fine-tune your designs, optimize your ejection delays, and achieve more successful flights. So, grab your altimeter, launch your rocket, and unlock the true potential of your simulations.

Credits:

I'd like to thank our product developer, Martin Jay McKee, for the help he provided in writing this article, particularly going over my math to make sure that everything was technically correct. When I started this article, I asked an A.I. chatbot to create the spreadsheet for me, and it got it completely wrong. It was a learning opportunity from the perspective that if you use A.I. to solve problems, you sort of need some perspective on what the correct answer should be.

I'm still for using A.I. in rocketry, as it was helpful to me to wrap my head around the problem, and come up with ideas to explore further. You should know that this was written by a real human.

About The Author:

Tim Van Milligan (a.k.a. "Mr. Rocket") is a real rocket scientist who likes helping out other rocketeers. He is an avid rocketry competitor and is Level 3 high power certified. He is often asked what is the biggest rocket he's ever launched. His answer is that before he started writing articles and books about rocketry, he worked on the Delta II rocket that launched satellites into orbit. He has a B.S. in Aeronautical Engineering from Embry-Riddle Aeronautical University in Daytona Beach, Florida, and has worked toward an M.S. in Space Technology from the Florida Institute of Technology in Melbourne, Florida. Currently, he is the owner of Apogee Components (http://www.apogeerockets.com) and also the author of the books: Model Rocket Design and Construction, 69 Simple Science Fair Projects with Model Rockets: Aeronautics and publisher of the "Peak-of-Flight" newsletter, a FREE ezine newsletter about model rockets. You can email him by using the contact form at https://www.apogeerockets.com/Contact.

SUBMITTING ARTICLES TO APOGEE

We are always looking for quality articles to publish in the *Peak-of-Flight* newsletter. Please submit the “idea” first before you write your article. It will need to be approved first.

When you have an idea for an article you'd like to submit, please use our contact form at https://www.apogeerockets.com/Contact. After review, we will be able to tell you if your article idea will be appropriate for our publication.

Always include your name, address, and contact information with all submissions. Including best contact information allows us to conduct correspondence faster. If you have questions about the current disposition of a submission, contact the editor via email or phone.

CONTENT WE ARE LOOKING FOR

We prefer articles that have at least one photo or diagram for every 500 words of text. Total article length should be between 2000-4000 words and no shorter than 1750 words. Articles of a “how-to” nature are preferred (though other types of articles will be considered) and can be on any rocketry topic: design, construction, manufacture, decoration, contest organization, etc. Both model rocket and high-power rocket articles are accepted.

CONTENT WE ARE NOT LOOKING FOR

We don't publish articles like "launch reports." They are nice to read, but if you don't learn anything new from them, then they can get boring pretty quick... Example: "Bob flew a blue rocket on a H120 motor for his certification flight." As mentioned above, we’re looking for articles that have an educational component to them, which is why we like “how-to” articles.

You can see what articles and topics we’ve published before at: https://www.apogeerockets.com/Peak-of-Flight?pof_list=archives&m=education. You might use this list to give you an idea or two for your topic.

Here are some of the common articles that we reject all the time, because we’ve published on these topics before:

- How to get a L1, L2, or L3 Cert
- Building cheap rockets and equipment (pads & controllers)
- How to 3D print parts, or a Rocket Kit
- How to Build a cheap Rocket Kit
- Getting Back Into Rocketry After a Long Hiatus

ARTICLE & IMAGES SUBMISSION

Articles may be submitted by emailing them to the editor. Article text can be provided in any standard word processor format, or as plain-text. Graphics should be sent in either a vector format (Adobe Illustrator, SVG, etc.) or a raster format (such as jpg or png) with a width of at least 600 pixels for single column images or 1200 pixels for two-column images. It is preferable for images to be simple enough to be readable in a two-column layout, but special layouts can be used.

Send the images separately via email as well as show where they go by placing them in the word processor document.

ACCEPTANCE

Submitted articles will be evaluated against a rubric (available here on our website). All articles will be evaluated and the results will be sent to the author. In the evaluation process, our goal is to ensure the quality of the content in *Peak-of-Flight*, but we want to publish your article! Resubmission of articles that do not meet the required standard are heavily encouraged.

ORIGINALITY

All articles submitted to Peak-of-Flight must not run in another publication before inclusion in the *POF* newsletter, but it may be based on another work such as a prior article, R&D report, etc. After we have published and paid for an article, you are free to submit them to other publications.

RATES

Apogee Components offers **$300** for a quality-written article over 2,000 words in length. Payment is pro-rated for shorter articles.

WHERE WILL IT APPEAR?

These articles will mainly be published in our free newsletter, *Peak-of-Flight*. Occasionally some of the higher-quality articles could potentially appear in one of Tim Van Milligan's books that he publishes from time to time.

Apogee Components, Inc.
4960 Northpark Dr.
Colorado Springs, CO 80918
719-535-9335

Your Success Is Our Mission!

www.ApogeeRockets.com