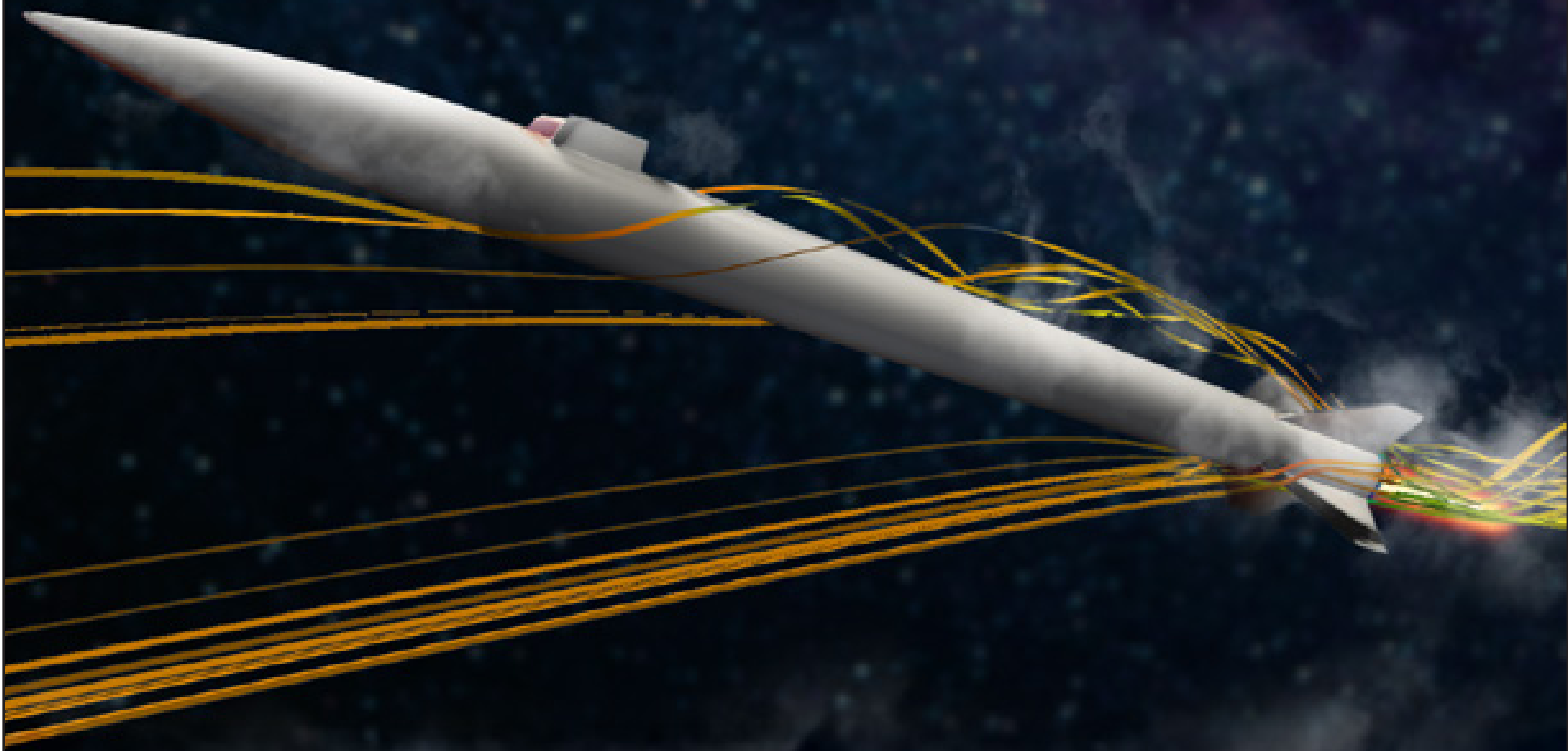
Issue 641 / December 17th, 2024

PEAK OF FLIGHT NEWSLETTER

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

Computational Fluid Dynamics in Model Rocketry



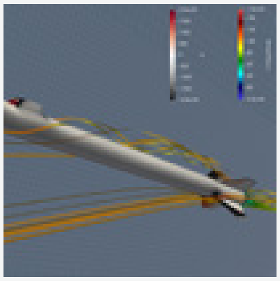
PEAK OF FLIGHT NEWSLETTER

Issue 641 / December 17th, 2024

COVER PHOTO

Fluid Dynamics Represented on a Model Rocket

FEATURED ARTICLES



A Guide to Computational Fluid Dynamics for Model Rocketry

by Ken Karbon

An intro to the science of Computational Fluid Dynamics (CFD) in Model Rocketry

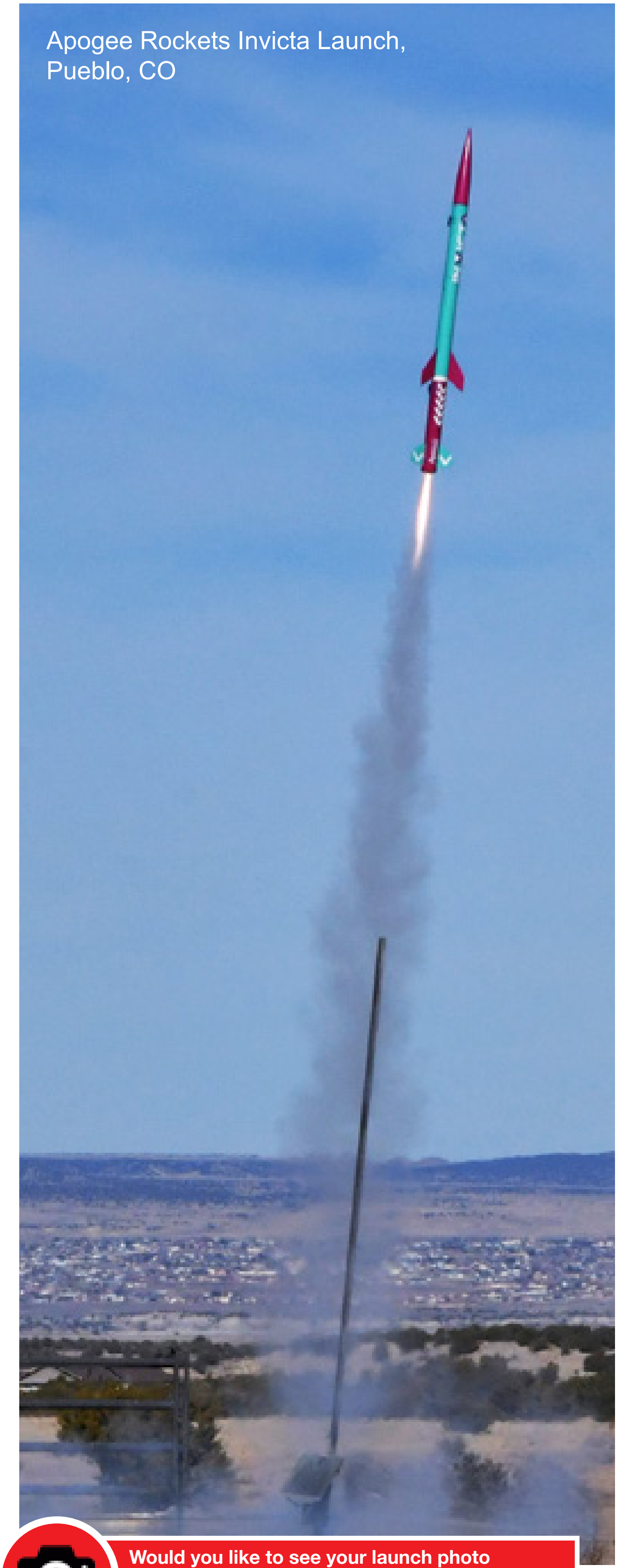
Apogee Components, Inc.
4960 Northpark Dr.
Colorado Springs, CO 80918
1-719-535-9335
www.ApogeeRockets.com

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

Apogee Rockets Invicta Launch, Pueblo, CO



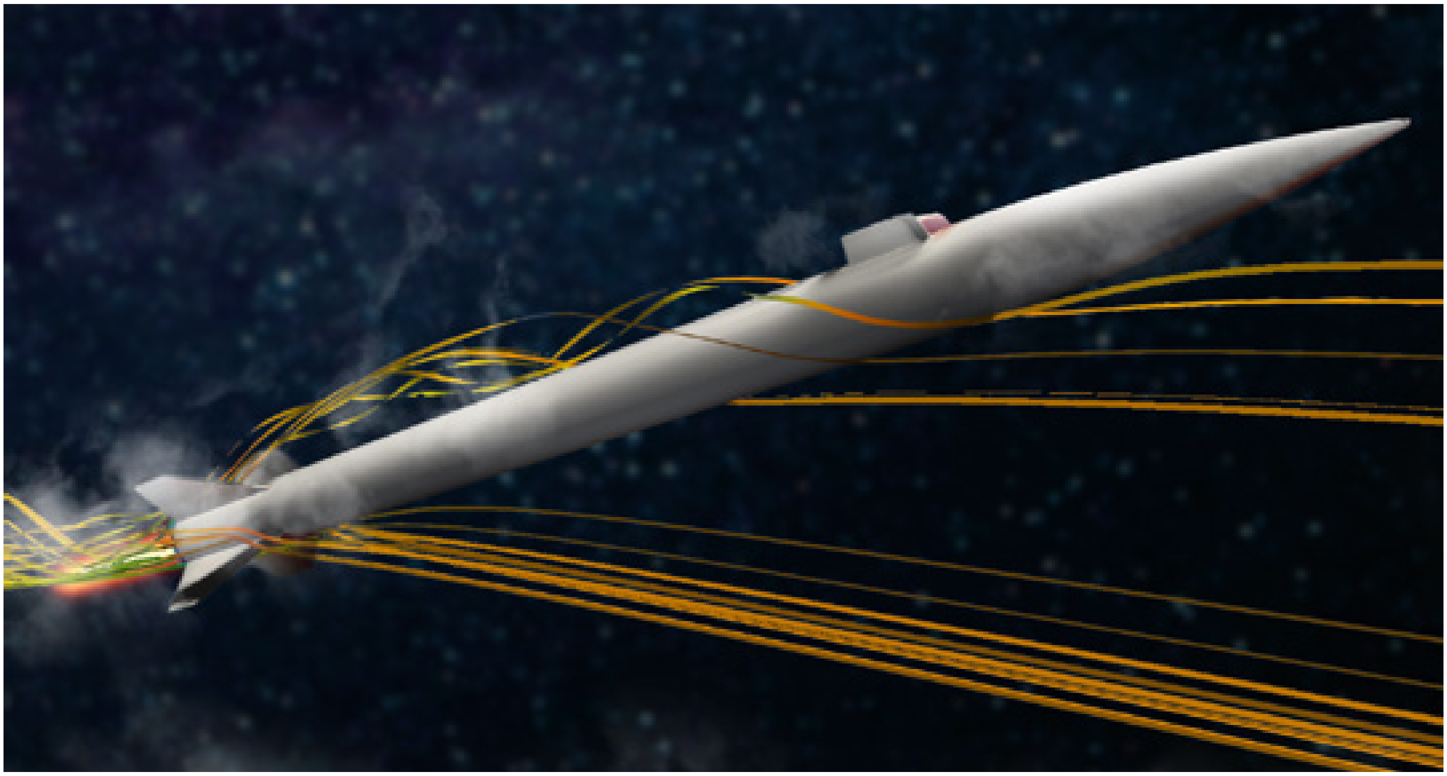
Would you like to see your launch photo featured in the *Peak-of-Flight* newsletter? Submit your photo at apogeerockets.com.

PEAK OF FLIGHT NEWSLETTER

Computational Fluid Dynamics
in Model Rocketry

Issue 641 / December 17th, 2024

www.ApogeeRockets.com



A Guide to Computational Fluid Dynamics for Model Rocketry

by Ken Karbon

Computational Fluid Dynamics (CFD) is the art and science of using numerical methods and computer data structures to solve fluid flow problems. CFD analysis falls into the broad category of Computer Aided Engineering (CAE), along with Finite Element Analysis (FEA), multibody dynamics, and others. People can generally understand the concept of structural analysis with FEA for example, since everybody has experience with bending and breaking things. CFD is often more difficult to grasp however, since it is not the object being calculated and analyzed, but rather the fluid around it. In real life, air (a fluid) is invisible and hard to visualize, adding to the mystique.

CFD solves the Navier-Stokes equations, which are equivalent to F = ma for fluids. These equations are unsteady, nonlinear, partial differential equations which are impossible to solve directly except for a few idealized cases. Instead, CFD breaks the problem into millions of small, algebraic equations across a "mesh." The equations are solved simultaneously & iteratively to convergence.

For rocket aerodynamics, we are interested in computing the pressure and velocity fields around the rocket and the resulting forces and moments that create drag, lift, and the center of pressure. Multiple simulations show how these quantities change due to changes in speed or angle of attack. Perhaps best of all, CFD data can be visualized in many ways like vectors, contours, and streamlines which aid in understanding.

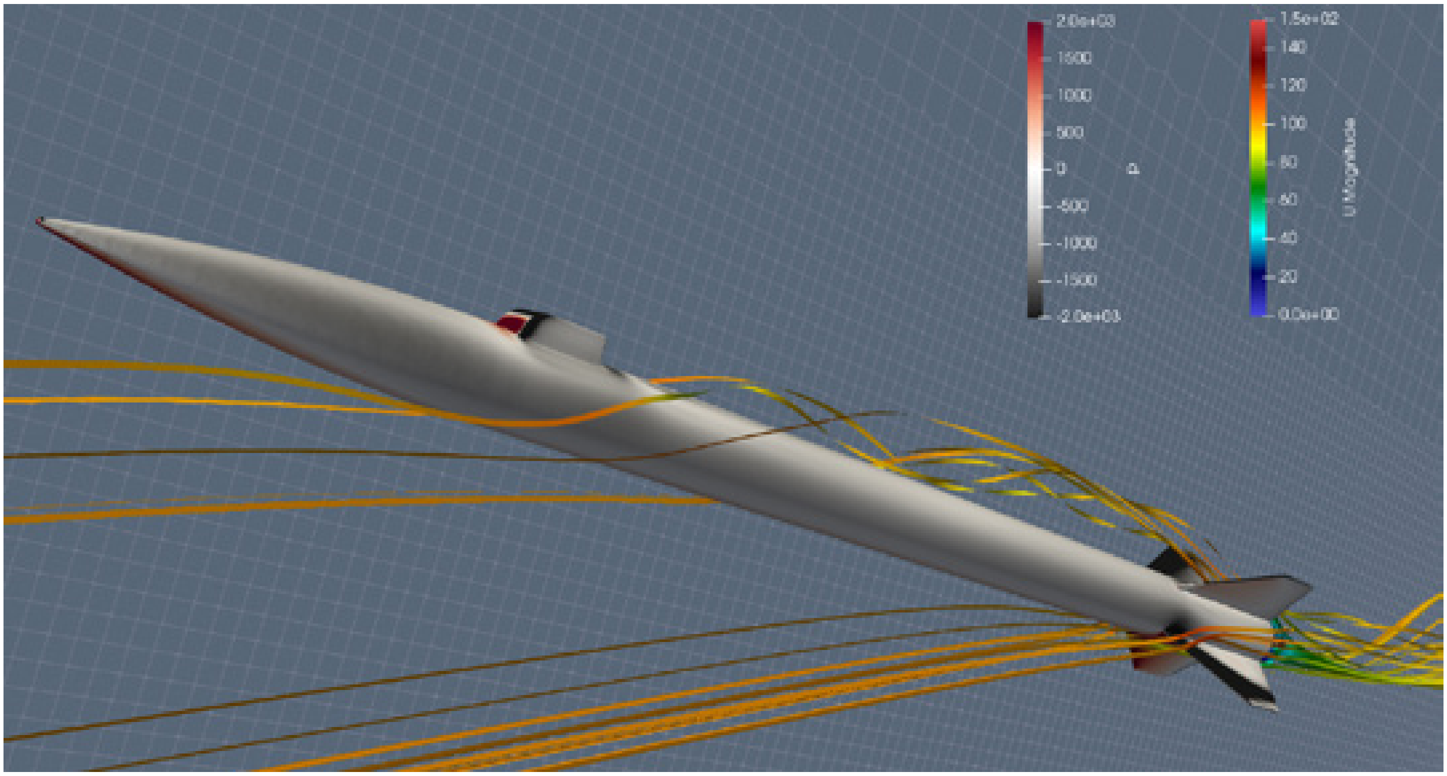


Figure 1: Post-Processing of CFD Simulation Results

Figure 1. illustrates some of the CFD concepts and capabilities to be covered in this article, all in one post-processing image. A model HPR rocket is flying at Mach 0.3 and an angle of attack of 16 degrees. The model includes an arbitrary geometry (camera shroud) that does not require a unique calculation of its aerodynamics. In the background is the computational mesh. The rocket is colored by static pressure, which when integrated, contribute to the total forces and moments on the body. Streamlines of velocity visualize the airflow direction and magnitude and highlight rotational flow and base wake.

In industry, CFD jobs usually require masters and PhD degrees and years of experience before an engineer is considered proficient. Commercial software licenses from Siemens, Ansys, Dassault and others are purchased for millions of dollars annually. High-end computer workstations with lots of memory and powerful graphics are used for pre- and post-processing. Calculations are performed on clusters of hundreds to thousands of CPUs (These used to be called "super computers," but are much more ubiquitous now.) So, this article can only scratch the surface of CFD, but some recent developments have made CFD more accessible and practical to hobbyists.

Open-Source CFD software

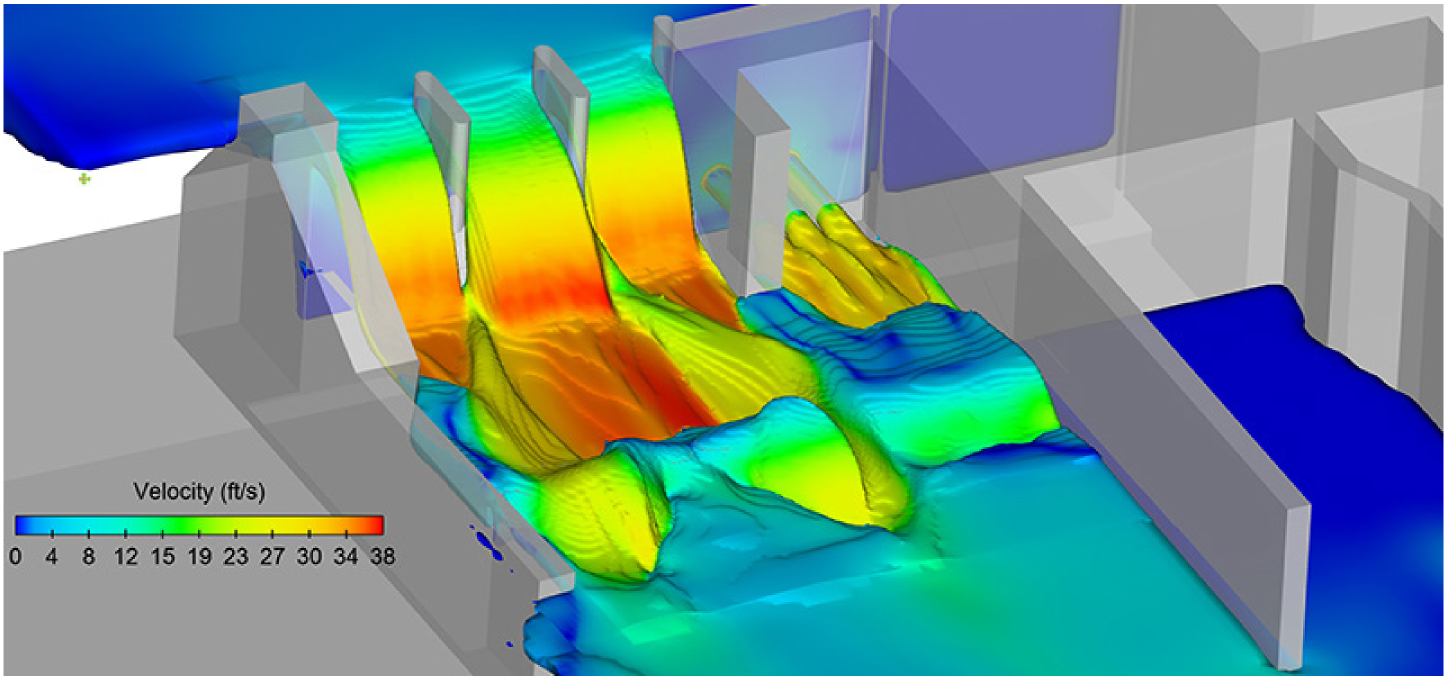
Open-source software is licensed to give users the rights to use, study, change, and distribute the software and its source code for any purpose (for a fee or gratis). The software is usually developed publicly and collaboratively. There are pros and cons to this approach, but open-source software usage is growing tremendously, with an estimated value of 8.8 trillion dollars to the companies that use it [1].

The oddly-named OpenFOAM (Open Field Operation and Manipulation) is the most notable open-source CFD code. Established in 2004, it is now widely accepted for aerodynamic analysis and many other industrial problems. One downside is that the default interface is merely a set of text files (called dictionary or "dict" files) that the user must edit and manipulate. Building a CFD simulation is not intuitive, and the learning curve is steep. To get past this, engineering consulting companies developed a market of offering custom GUIs, training, and support for OpenFOAM.

There are many 3rd party helper software for OpenFOAM. These typically have license fees, but the costs are far less than a proprietary CFD package. I was looking for such a software for personal use for some time, ideally at low cost or even free of cost. Then, I saw David Carter's presentations at NARCON 2023 and NARCON 2024 discussing the model rocket capabilities in FreeCAD. FreeCAD is open-source CAD software with the ability to add "Workbenches" that perform specific tasks. David is a software developer, and he created the Rocket Workbench, which consists of easy-to-use functions that build 3D, parametric, CAD models of rocket parts like nose cones, fin cans, rail lugs, and transitions. These models are suitable for 3D printing.

David also introduced the CfdOF Workbench. This workbench provides a front-end GUI to guide the user through the entire CFD process with OpenFOAM, including robust best practices built in. This was exactly what I was looking for and a breakthrough for hobbyists in my view. No longer restricted to pricey, proprietary software at the workplace, full-featured, Computational Fluid Dynamics is feasible for individuals on a home computer.

CFD Software showing a model of a flowing fluid in a dam



CFD for Model Rocket Aerodynamics

Using CFD for hobby rocketry is not new, of course, but there were some limitations in the past. Rocketeers who knew how to execute CAE needed access to employer resources to perform simulations of their models. Apogee sold a product called “AeroCFD,” but it was restricted to 2D axisymmetric rockets and Euler flows. Tim Van Milligan did a launch lug drag R&D project using “Flow Solver” within Autodesk. A common downside to CFD packaged with CAD programs is that the CFD solver will be a “light” version with reduced physics, outputs, or model size. Tim encountered some of these issues.

OpenFOAM is industrial strength. The full Navier-Stokes equations are solved, so there are no limitations to the fluid regime. Nearly all fluid flow phenomena are supported, including turbulence, supersonic, free-surface, heat transfer, and more. The only limitation is the amount of computing power you can throw at it.

The traditional aerodynamic simulations (like Estes TR-11) are semi-empirical models fit to mathematical curves for ease of computation. These models are up to 60 years old. Components are analyzed separately, and then superimposed and added together to come up with the total rocket value. Additional models are needed to account for the interference between components. Components consist of basic fins, bodies, transitions, and nose cones, with some additional variations of these shapes. The more your model rocket deviates from the shapes used to build the equations, the less accurate the simulation.

The Barrowman equations for Center of Pressure, while simple and fast, have a set of assumptions that are often overlooked and violated, including:

- Potential flow, i.e. no vortices or friction
- Point of the nose is sharp
- Fins are thin flat plates
- Angle of attack is very near zero
- Flow is steady-state and subsonic
- The rocket is axially symmetric

CFD does not have these constraints. The aerodynamics are solved holistically from first principles. Any shape you can design in CAD can be simulated with the correct fluid mechanics. More accurate values for CD, CNa, and CP can be generated, along with “seeing” the airflow. It is like a wind tunnel, but more attainable for rocketeers.

Very few hobbyists have access to quality wind tunnels that record accurate measurements. (Tunnels made from cardboard and a box fan are good for simple airflow demonstrations, but not much else.) The physical construction of the tunnel limits

the size of model and flow conditions to be tested. On the other hand, CFD is available in various forms, and it is easy to adjust the virtual set up to different models or flight regimes. The CFD mathematical techniques I will show here are the same as those used in industry and have been refined and validated over many years. As computing gets bigger, faster, and more innovative, so do the CFD applications. GPU solvers, machine-learning, and artificial intelligence are all making impacts in the CAE world.

Now for the disclaimers. Running CFD simulations are not trivial and are easy to screw up. Solutions are sensitive to choice of physics models and mesh size. The user really should have experience in CAE, scientific computing, and fluid mechanics. You need the ability to distinguish good models from bad, to interpret results, and to fix problems.

Modern software and automated workflows with best practices are certainly making CFD setup easier. The initial startup hurdles are removed and the “scary factor” is reduced. Even though I have many years of CFD experience and used OpenFOAM before, I still spent several weeks of trial and error within FreeCAD and CfdOF to figure out the best approaches and to confidently believe in the results. Hopefully, the information I present in this article will help others avoid the initial pitfalls.

Unlike RockSim and other rocketry computer tools, the CFD results are not calculated instantaneously. A sweep of speed or angle of attack requires many individual computer runs. My solutions take a few hours to solve per configuration on a basic desktop computer with these specs:

- OS: Microsoft Windows 10 Home
- CPU: AMD Ryzen 3 PRO 4350G, 3800 MHz, 4 Core(s)
- 16 GB RAM
- NVIDIA Quadro P400 graphics card

The only slightly “upscale” component I have is the dedicated Quadro P400 graphics card. It is a low-end card optimized for CAD (not gaming) applications, which is a big help in visualizing graphics-heavy CAD and CAE data. If you are a computing pro with a “threadripper” workstation, you can get results much faster.

Figure 2 shows a typical CFD process from start to finish. The following is not a tutorial in FreeCAD or CfdOF, but rather a discussion on the salient aspects of each step that applies to any software. The theory and math are extremely complicated, so I will try to keep things in practical terms. My background is in automative aerodynamics, so I may have an incorrect detail here or there when it comes to aircraft and rocket CFD simulations. The overall concepts remain the same, however.

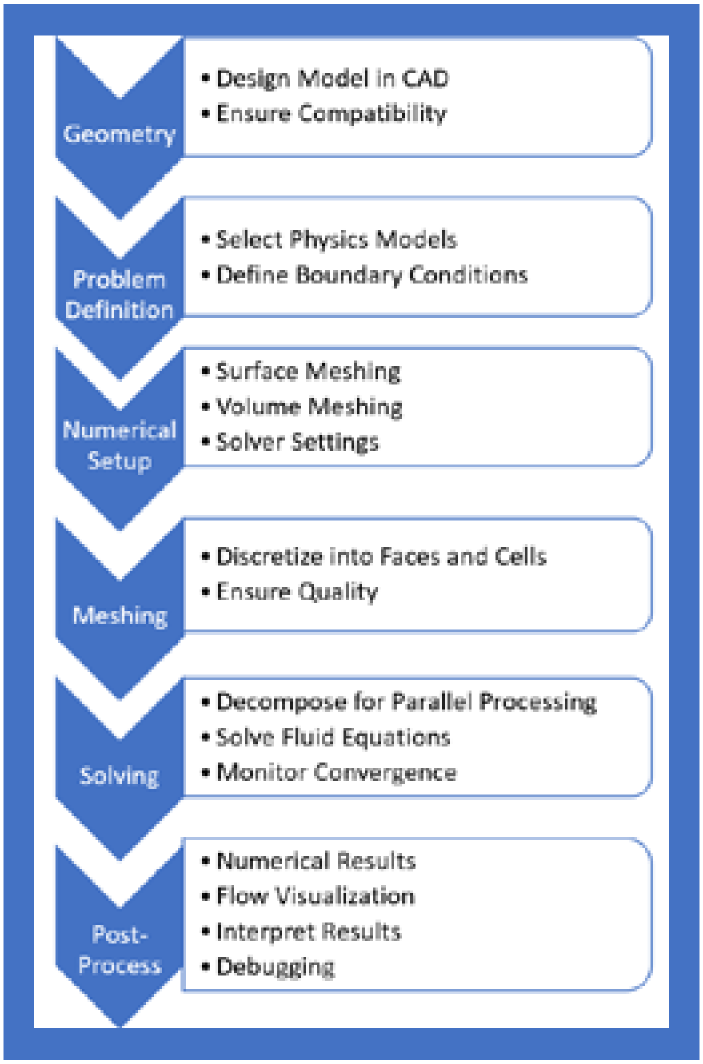


Figure 2: The Six Step CFD Workflow Diagram

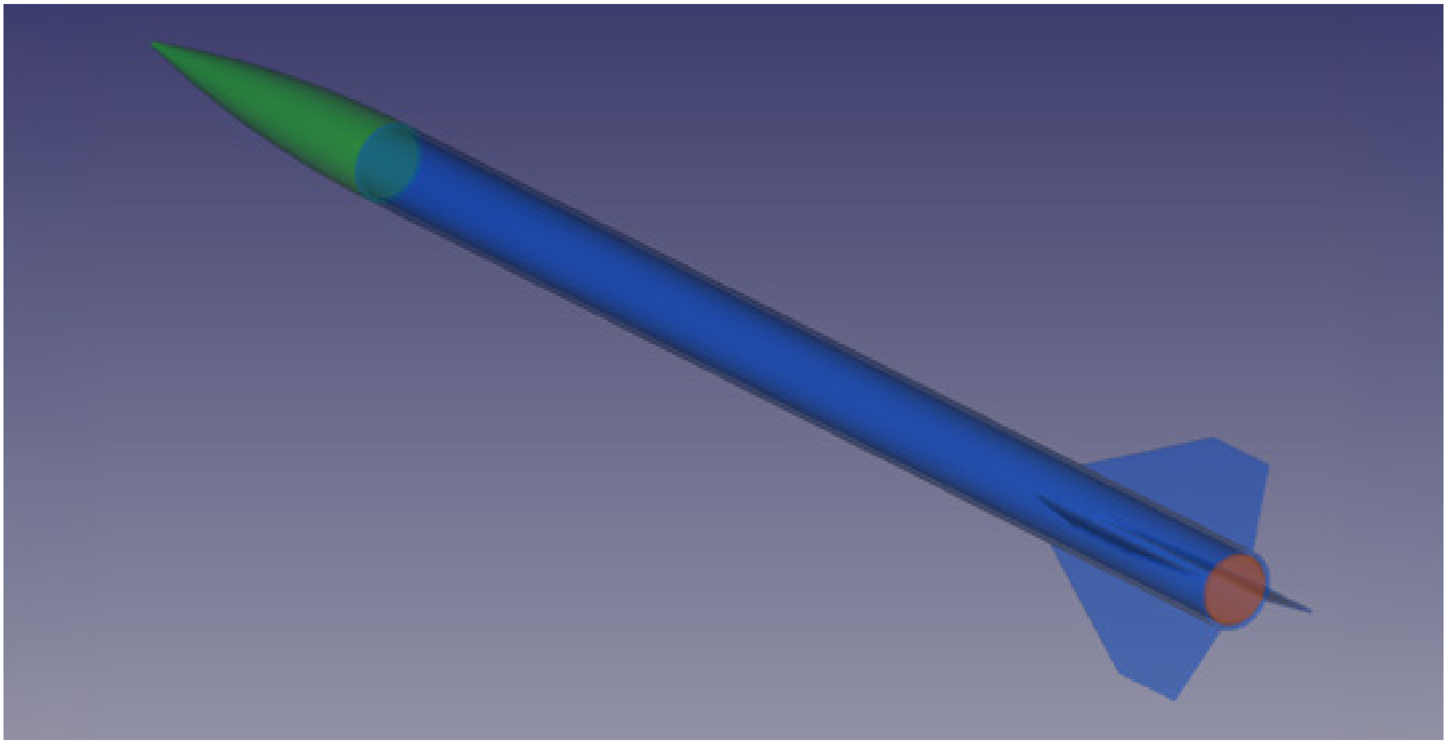


Figure 3: A Simple Rocket CAD Model

Geometry

The process starts with designing the rocket as a 3D CAD model. The goal is to produce a "watertight" geometry with no overlaps, cracks, or duplicate surfaces. For aerodynamic studies, only the outside of the rocket is important. Hence, The internal volumes of the rocket are dead space, and we do not need nor want CAD mesh or calculations in those spaces. In industry, CAD models are complicated and focused on manufacturing needs and are usually not "clean" enough for CAE analysis. Cleaning geometry has been the bane of CAE for decades, and there are many powerful techniques and software have been developed to make it easier. However, it is still not perfect. The CAE engineer's skill and judgement are needed to heal the geometry and ensure that it is suitable for CFD analysis. This is done manually on the CAD entities, with automated "shrink-wrapping," or via other utilities in the CFD software itself. The hobby rocket geometries we build are generally simple, so it is easy to keep them clean and watertight as they are being designed.

It is best to work in a full-featured CAD software like FreeCAD, Fusion, or the myriad of others in the marketplace. These solid modeling software are powerful for creating, editing, & exporting precise curves and surfaces as Non-Uniform Rational B-Spline (NURBS) formats like IGES or STEP that can be read by CFD preprocessors. FreeCAD is especially convenient in that the CAD and CFD are all in one package, meaning that there is no need to transfer the data between software. FreeCAD is fully parametric, meaning changes in the CAD model automatically propagate into the CfdOF setup as well.

Figure 3 is a simple, 3-inch rocket for CFD studies that I made in the FreeCAD Rocket Workbench. It consists of just 3 solid objects: Nose cone, fin can (albeit a very long one), and bulkhead at the rear to seal off the internal volume. The Rocket Workbench is not only great at designing parts for 3D printing, but it also makes quick work of building CFD models!

You may be tempted to build a rocket in RockSim and then export it as a 3D model such as .obj or .vrml for direct use in CFD. While this is technically feasible and a great goal for the future, there are problems with this approach today. Those file formats are not solids, nor NURBS. They are simply triangles. Thousands of them. The triangle facets are good for visual rendering and 3D printing, but not very good for computational models and analysis. The triangles have extremely high aspect ratios, are difficult to edit, and lack connectivity between adjacent components.

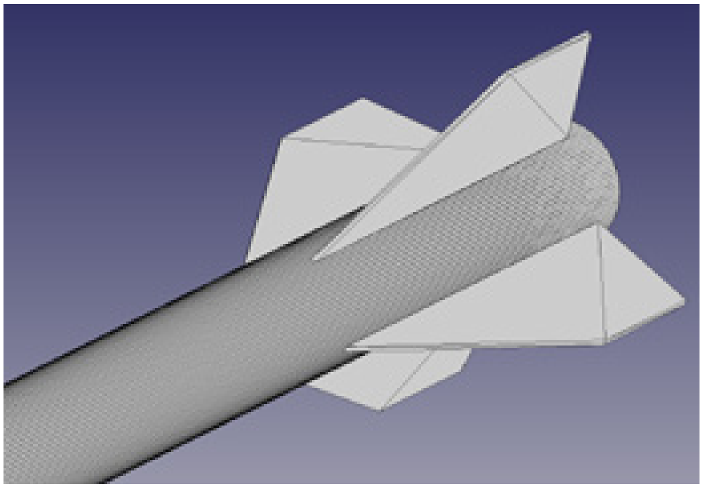


Figure 4: .obj File Format Exported from RockSim

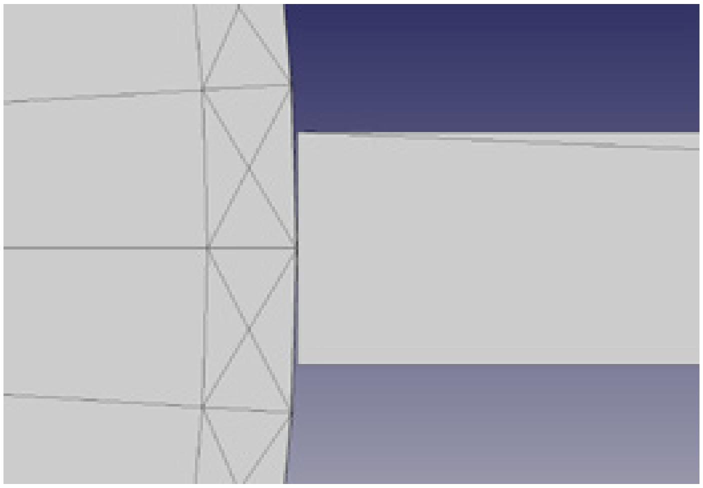


Figure 5: Fin to Body Interface in RockSim .obj Export

Figure 4 shows a typical .obj export from RockSim. The rocket body tube is made of extremely long and skinny triangles and just 10 triangles define each fin. These would lead to very inaccurate computations if used directly by the CFD solver. As we'll see later in this article, computational meshes look much different.

Even as a basis for the shape of the rocket, the faceted model also presents problems. Figure 5 is an end view of the rocket with a fin butting against the body tube. The fin joins the body tube merely along a tangent line, creating tiny, unrealistic air gaps along the remainder of the fin root. CFD may struggle to mesh and solve these very small angles and crash the process.

With effort, the triangles can be converted, massaged, and forced into compliance for CFD. However, I find it much easier to work on native CAD entities at this time. The above issues are easily resolved in CAD software before attempting the CFD process. In Figure 6, the fin roots are sunk into the body tube to eliminate the tangent contact. The Rocket Workbench in FreeCAD does this automatically when creating a fin can.

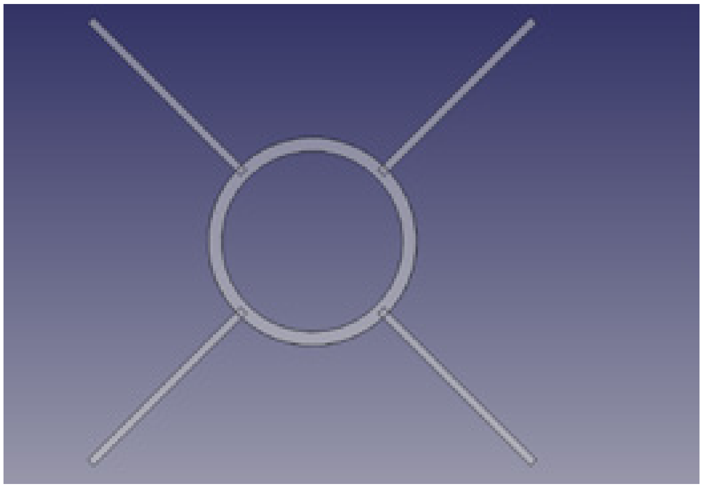


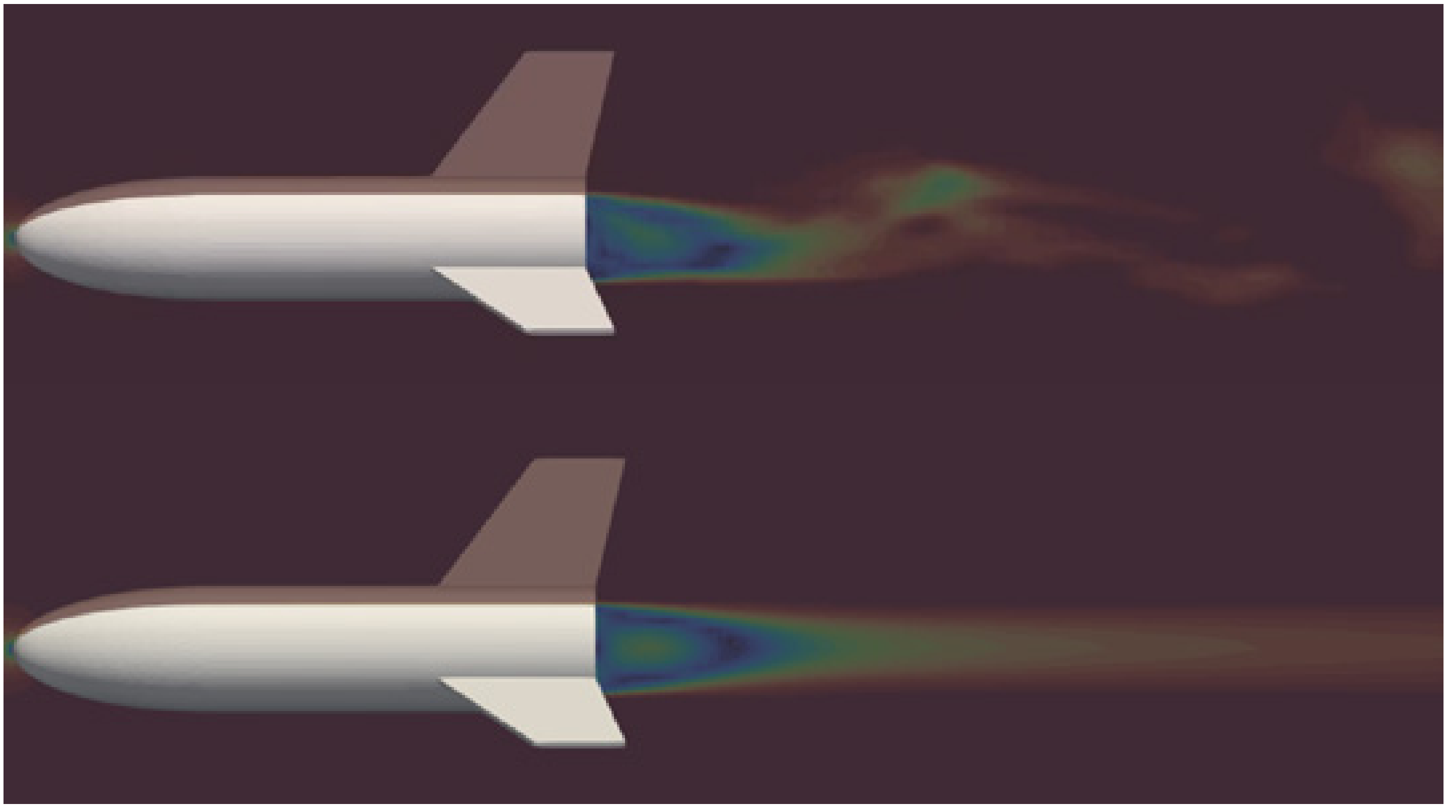
Figure 6: Embedded Fins in CAD Model

Physics Models

As the famous automotive engineer Charles Kettering once said, "A problem well stated is a problem half solved." The regime in which the rocket is flying determines the physics models used. For subsonic flights less than Mach 0.3, an incompressible solver is a good assumption, meaning the air density remains constant. For supersonic flights, where air density changes across shock waves, a compressible solver would be used.

Aerodynamics are inherently transient, where air quantities like pressure and velocity can change over time and space. This is especially true in the wake of the rocket where there is some flow separation and rotating vortices. These changes occur at very small intervals, and require calculation time steps on order of 1.0E-06 seconds. Many steps are needed to resolve a couple periods of oscillating flow behavior. This comes with numerical stability concerns and tremendous computational cost that even organizations with large servers struggle with.

There is a method of averaging called the Reynolds Averaged Navier Stokes equations (RANS) that march towards a steady-state solution with just a fraction of the computing time. Our rockets are usually streamlined, the bulk flow develops quickly, and the transient behavior is limited to small regions. Thus, a steady-state solver is a reasonable choice for most hobby cases on a home computer. Figure 7 shows a slice of velocity contours in the wake behind the rocket. On the top is a snapshot in time of a transient simulation which captures the wake fluctuating back and forth. On the bottom is the steady-state "average" solution with RANS which remains fairly constant once the solution is converged.



Rocket Base Wake. Transient Simulation Snapshot (Top) and Steady Simulation (Bottom)

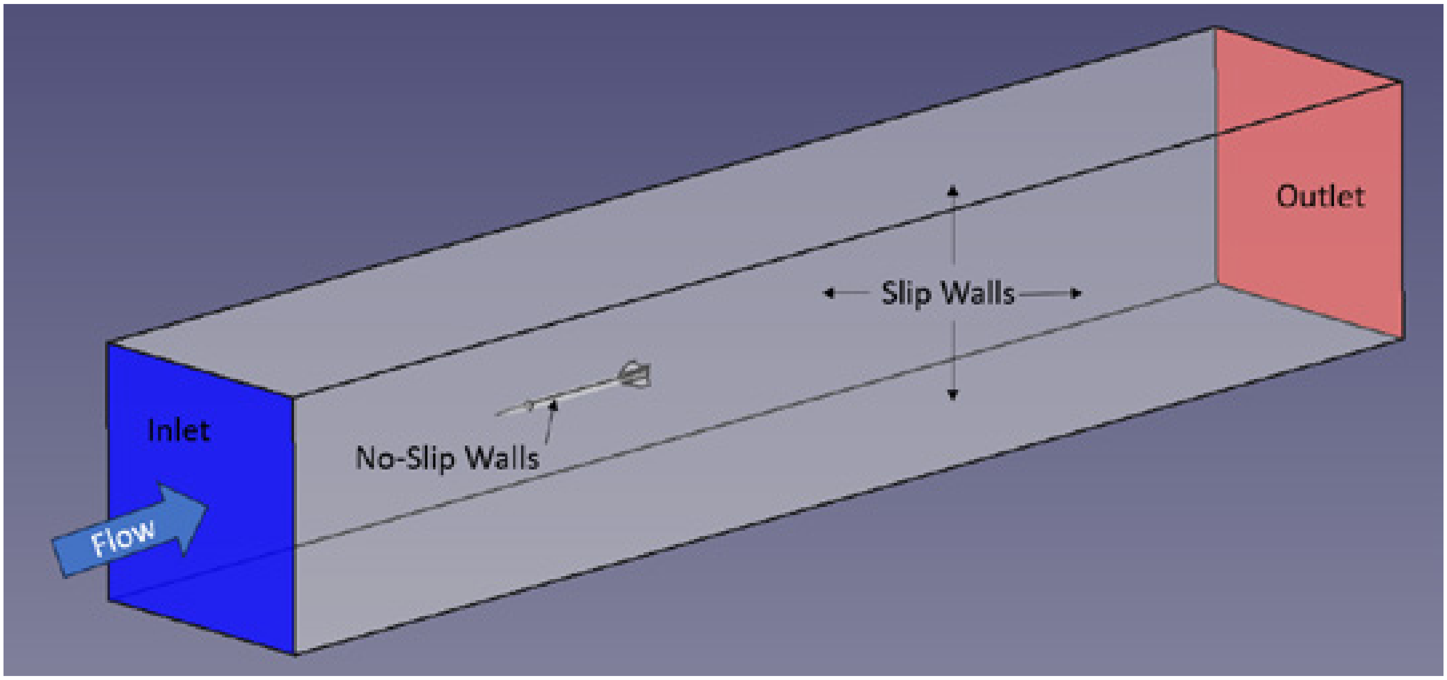


Figure 8: Boundary Conditions

Ideally, we want our rockets to have laminar flow to minimize friction drag. On sport models, this probably does not occur due to imperfections or protrusions quickly tripping the flow from laminar to turbulent. CFD is not great at transitioning from laminar to turbulence calculations, so selecting all turbulent flow is a good choice. Modeling turbulence remains an area of research, and many formulations exist. K-epsilon and k-omega are a couple well-known ones for external aerodynamics.

Boundary Conditions

In numerical simulation, boundary conditions refer to extents of the model where we already know the answer. They do not have to be calculated. Since CFD is like a virtual wind tunnel, we can set up the simulation like a wind tunnel facility for incompressible flows. The rocket is suspended in a long box (or cylinder) with air flowing past it. This is the “computational domain.” There is an inlet to the box where we specify the speed of the oncoming air. There is an outlet to the box, where the airflow returns to its original condition after being disturbed by the rocket in its path. The sides of the box are slip walls, which allow the air to move tangentially across their faces, but prevents air from moving normal to their faces. Lastly, the rocket surface is a no-slip wall, where the air flow can “stick” to the rocket and come to a stop with zero velocity. Everything else, namely the air space inside the box, must be calculated. Figure 8 shows the setup.

Figure 9: Computational Domain Length Guideline

The setup should mimic the rocket flying in free air. The size of the domain needs to be sufficiently large such that the inlet, outlet, and slip walls are far away and do not influence the airflow near the rocket. Figures 9 and Figure 10 give approximate sizes that have worked well for me and minimize the Bernoulli effect of constricting the domain.

Compressible, or high Mach Number flows use “Far-Field” boundary conditions to avoid shock wave interaction with the bounds of the domain. The air is assumed to be at ambient conditions in the far field, including temperature. The size of this box is usually much larger than inlet-outlet tunnels.

The mesh spacing inside the domain dictate the computing effort and accuracy, so they must be chosen wisely. More on that later as the remainder of the CFD process is covered in part 2 of this series.

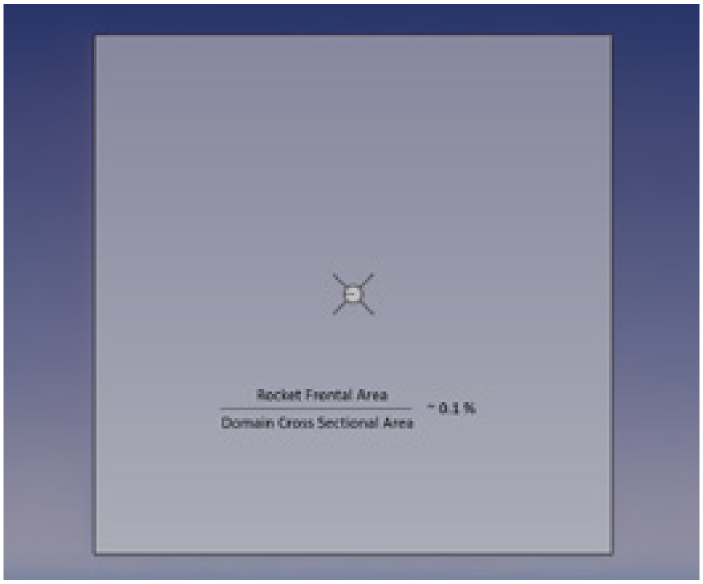


Figure 10: Computational Domain Cross Section Guideline

About the Author

Ken Karbon is a rocketeer from Michigan. He is a retired engineer from the auto industry where he specialized in CFD and aerodynamics.

PEAK OF FLIGHT NEWSLETTER

Issue 641 / December 17th, 2024

www.ApogeeRockets.com

Article Submission Guidelines

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