15 May 2022

Updates And Improvements To The Satellite Drag Coefficient Response Surface Modeling Toolkit

Phillip Logan Sheridan

Smriti Nandan Paul
Missouri University of Science and Technology, smritinandan.paul@mst.edu

Guillermo Avendaño-Franco

Piyush M. Mehta

Recommended Citation

P. L. Sheridan et al., "Updates And Improvements To The Satellite Drag Coefficient Response Surface Modeling Toolkit," *Advances in Space Research*, vol. 69, no. 10, pp. 3828 - 3846, Elsevier, May 2022.
The definitive version is available at https://doi.org/10.1016/j.asr.2022.02.044

Available online at www.sciencedirect.com

ScienceDirect

Advances in Space Research 69 (2022) 3828–3846

ADVANCES IN SPACE RESEARCH
(a COSPAR publication)

www.elsevier.com/locate/asr

Updates and improvements to the satellite drag coefficient Response Surface Modeling toolkit

Phillip Logan Sheridan [a,*], Smriti Nandan Paul [a], Guillermo Avendaño-Franco [b]
Piyush M. Mehta [a]

[a] *West Virginia University, ASSIST Laboratory, West Virginia University, Morgantown 26506, United States*
[b] *West Virginia University, Research Computing, West Virginia University, Morgantown 26506, United States*

Received 20 November 2021; received in revised form 22 February 2022; accepted 23 February 2022
Available online 4 March 2022

Abstract

For satellites in the Low Earth Orbit (LEO) region, the drag coefficient is a primary source of uncertainty for orbit determination and prediction. Researchers at the Los Alamos National Laboratory (LANL) have created the so-called *Response Surface Modeling (RSM)* toolkit to provide the community with a resource for simulating and modeling satellite drag coefficients for satellites with complex geometries (modeled using triangulated facets) in the free molecular flow (FMF) regime. The toolkit fits an interpolation surface using non-parametric Gaussian Process Regression (GPR) over drag coefficient data computed using the numerical Test Particle Monte Carlo (TPMC) method. The fitted response surface provides a substantial computational benefit over numerical approaches for calculating drag coefficients. In this work, the RSM toolkit is further developed into a versatile software with extended capabilities. The capabilities are specifically expanded to include uncertainty quantification and adaptation for automatic development of regression models for satellites with non-stationary components (e.g. rotating solar panels). Furthermore, the toolkit uses Python 3.x and C programming languages to provide an open source software package with a OSI approved GPL license. To assist the end user, the new RSM toolkit has been developed to have a user-friendly installation process and is provided with extensive documentation. The analysis of two different conceptual satellites is performed during this work: a simple cube and a CubeSat consisting of a simple cube body with 2 rotating solar panels. During the creation of the regression model for each satellite for different atmospheric species, it is found that the cube's minimum Root Mean Squared Error (RMSE) is 0.00211 and the maximum RMSE is 0.00350. The CubeSat has a minimum RMSE of 0.00304 and the maximum is 0.00498. These results are overall conducive of a well performing regression model.

Keywords: Gas-surface interaction; Response surface modeling; Drag modeling; Satellite drag coefficient

1. Introduction

The near-Earth space environment has experienced expeditious object population growth in the recent past because of increased participation from private companies. According to IEEE, the Federal Communications Commission (FCC) has approved SpaceX to launch a total of nearly 12,800 satellites for its Starlink constellation. To date, SpaceX has launched 1,300 of these satellites (SpaceNews, 2020). Satellite constellations for the internet and other consumer services are becoming more and more popular. Companies like OneWeb, Telesat, and Kuiper Systems LLC also plan to send satellites en masse to orbit. Between these four companies, 46,100 satellites are planned to be launched in the next decade (CNBC, 2020). With so many satellites in orbit, it is imperative that operators have

* Corresponding author.
E-mail addresses: pls0013@mix.wvu.edu (P.L. Sheridan), smritinandan.paul@mail.wvu.edu (S.N. Paul), gufranco@mail.wvu.edu (G. Avendaño-Franco), piyush.mehta@mail.wvu.edu (P.M. Mehta).

https://doi.org/10.1016/j.asr.2022.02.044
0273-1177/© 2022 COSPAR. Published by Elsevier B.V.

an accurate knowledge of the location of their assets at any given time, i.e., predict the orbital states. Furthermore, NASA states that, as of 2021, there are more than 27,000 pieces of orbital debris that are being tracked by the Department of Defense's global Space Surveillance Network (NASA, 2021). Most of these satellites and debris are going to (or already do) reside in the LEO region, where drag is the dominant source of orbit error resulting from inaccurate drag coefficient and atmospheric density modeling. The range of LEO used for this work involves the free molecular flow (FMF) regime, typically $>\sim 150$ km in altitude above the surface of the Earth and encompasses the high-altitude thermosphere. The thermosphere is the neutral part of the upper atmosphere that is generally dominated by atomic oxygen but significant changes in composition can occur depending on the altitude and solar conditions (Emmert, 2015). Improved drag modeling will help us make more informed decisions about collision avoidance maneuvers, thereby preventing a Kessler syndrome-like scenario (Kessler et al., 2010). The work presented here is targeted at improving modeling of the satellite drag coefficient. The focus is on the development of a drag coefficient modeling software that provides uncertainty estimates in a computationally efficient manner while being user-friendly and open source.

The theoretical model for calculating satellite drag is given by:

$$\overrightarrow{a}_{drag} = -\frac{1}{2}\rho\frac{C_D A}{m}v_{rel}\overrightarrow{v}_{rel} \tag{1}$$

where $\overrightarrow{a}_{drag}$ is the acceleration due to drag, ρ is the atmospheric density, C_D is the drag coefficient, A is the projected area of the satellite as seen by the flow, m is the mass of the satellite, $\overrightarrow{v}_{rel}$ is the velocity of the satellite with respect to the atmosphere, and v_{rel} is the magnitude of $\overrightarrow{v}_{rel}$. Error in calculating the acceleration due to drag comes from two main sources: density and drag coefficient. An important group of variables also worth noting is $\frac{m}{C_D A}$. Together, these variables make up the *ballistic coefficient*. The ballistic coefficient wraps the uncertainty of the satellite-specific variables into a singular variable of uncertainty.

1.1. Satellite drag coefficient

In the context of orbital dynamics, there are three ways to compute the drag coefficient (C_D): a physical C_D, a fitted C_D, and a fixed C_D (Moe et al., 2010). A fixed C_D is a constant value. The fitted C_D is estimated as part of an orbit determination process. It is specific to the atmospheric model used and includes the limitations of the atmospheric model. It also frequently absorbs other force model errors. Physical C_D is determined by the exchange of energy and momentum of the freestream atmospheric molecules with the spacecraft surface (Chambre and Schaaf, 2017). Throughout this work, the term C_D will refer to a physical C_D. The C_D is dependent on various parameters that include the free-stream velocity, temperature, and composition, surface temperature, satellite surface geometry, and gas-surface interactions (GSIs). The C_D has been shown to be most sensitive to GSI and strongly influenced by the spacecraft shape or geometry.

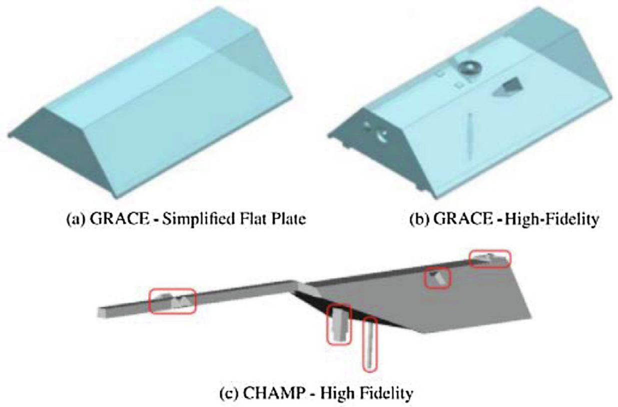
1.1.1. Satellite geometry definition

Geometry has been show to play a large role in the fidelity in estimating C_D for satellites (Pilinski et al., 2011; Pilinski et al., 2013; Mehta et al., 2017; March et al., 2019a; March et al., 2019b; Bernstein et al., 2020; March, 2021). Fig. 1 shows two satellites of interest: GRACE (Ramillien et al., 2004), the Gravity Recovery and Climate Experiment satellite, and CHAMP (Reigber et al., 2006), the Challenging Minisatellite Payload satellite. High-fidelity geometry models include components such as antenna, solar arrays, sensors, etc. This can be seen in the comparison of GRACE in Fig. 1 where GRACE is seen as a simplified flat plate versus its high-fidelity model which includes the miscellaneous components on the surface of the satellite. CHAMP is an example of a satellite with relatively large components, a better example as to why the fidelity of the model can have a significant impact on the C_D of the object.

1.1.2. Gas-surface interactions

GSI models describe the exchange of energy and momentum between atmospheric molecules and the surface of an object. The energy exchange is represented by adsorption modeling, while the momentum exchange is represented by reflection modeling. Two accommodation coefficients are used to define a GSI model: the tangential momentum accommodation coefficient (σ_t) and the energy accommodation coefficient (α). The energy accommodation coefficient is defined as:

$$\alpha = \frac{E_i - E_r}{E_i - E_s} \tag{2}$$


(a) GRACE - Simplified Flat Plate

(b) GRACE - High-Fidelity

(c) CHAMP - High Fidelity

Fig. 1. GRACE as a simplified flat plate (a) and as a high-fidelity model (b), CHAMP as a high-fidelity model. The red boxes are components that can not be simplified as a flat plate. (Mehta et al., 2017).

where E_i is the kinetic energy of an incident molecule, E_r is the kinetic energy of a reflected molecule, and E_s is the energy that a molecule would have after being re-emitted in thermal equilibrium with the surface. The other coefficient, σ_t, defines how the particle is reflected from the surface. If the tangential momentum coefficient is equal to one, then it is said that the reflection is diffuse. If the coefficient is equal to zero, then it is said that the reflection is specular. Any other values of σ_t is said to correspond to the quasi-specular reflection (Walker et al., 2014b). Fig. 2 is an illustration of the different reflections. Ultimately, the tangential momentum coefficient controls the scattering kernel for the particles in the software. The kernel that we use was developed by Lord (1991) and can be seen in Eq. 3.

$$P(v_{i_t} \rightarrow v_{r_t}) = \frac{1}{\sqrt{\pi\sigma_t(2-\sigma_t)}} e^{\frac{\left(-v_{r_t}-(1-\sigma_t)v_{i_t}\right)}{(\sigma_t(2-\sigma_t))}} \tag{3}$$

where v_{i_t} is the incident tangential velocity component, v_{r_t} is the reflected tangential velocity component, and $P(v_{i_t} \rightarrow v_{r_t})$ is the probability that the v_{i_t} is reflected as v_{r_t} (Walker et al., 2014b).

Together, these coefficients define the GSI modeling. Several GSI models have been developed to simulate these types of gas-surface interactions, including Maxwell's Model (Maxwell, 1879), Cercignani-Lampis-Lord (CLL) (Cercignani and Lampis, 1971), and Diffuse Reflection with Incomplete Accommodation (DRIA) (Pilinski et al., 2010; Moe et al., 2004; Sutton, 2009).

Adsorption models. Laboratory experiments began in the early 1900s to measure the reflection of molecules from surfaces. But, since the 1960s, satellite experiments have been performed using pressure gauges and mass spectrometers to reveal that satellite surfaces are covered by adsorbed gases which can vary given different altitudes (Moe and Moe, 1969). Moe et al. (1998), in the first figure of their paper, presents atomic oxygen surface coverage data from the mass spectrometer onboard NASA's OGO-6 satellite. The mass spectrometer data demonstrate that the surface coverage of oxygen is much higher at perigee (P) than at apogee (A). Adsorption occurs at lower altitudes and desorption occurs at higher altitudes (Moe and Moe, 1969). This realization has led to the necessity of creating different adsorption models of oxygen. While these measurements are not without errors (e.g. atomic oxygen recombination in old mass spectrometers), the hypothesis that adsorption is driven by the amount of atomic oxygen in the vicinity of the satellite has been widely accepted in the community.

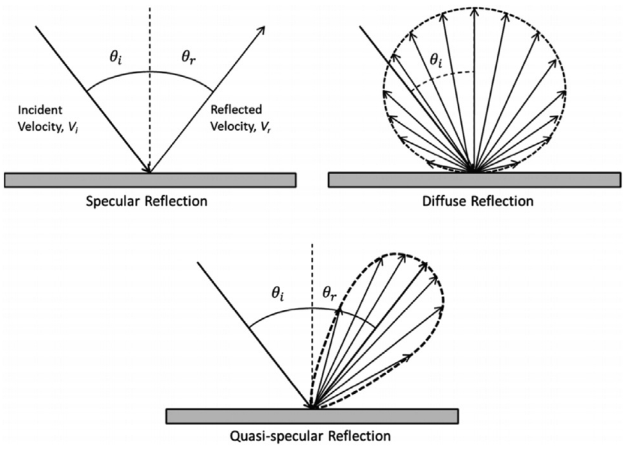


Fig. 2. Reflections of different gas-surface interactions (Walker et al., 2014b).

An adsorption isotherm is an empirical representation of the phenomenon governing the retention, release, or mobility of a substance from a fluid medium to a solid at a constant temperature. To date, there has been a total of fifteen isotherms developed (Foo and Hameed, 2009). Two of the most notable isotherms are the Freundlich (Freundlich, 1907) and Langmuir (Langmuir, 1916) isotherms, both of which are used in the development of the RSM software outlined later in this work.

Freundlich isotherm. The Freundlich isotherm is the first adsorption isotherm created in 1906 by Herbert Freundlich (Freundlich, 1907). It is a robust empirical model that can be applied to multi-layer adsorption, with non-uniform distribution of adsorption heat and affinities over a heterogeneous surface (Foo and Hameed, 2009). The following equation is the empirical model of the Freundlich isotherm:

$$\frac{x}{m} = kP^{\frac{1}{n}} \tag{4}$$

where x is the mass of adsorbate adsorbed, m is the mass of the adsorbent, P is the pressure of adsorbate, and k and n are empirical constants for each adsorbent-adsorbate pair at a given temperature.

Langmuir Isotherm. The Langmuir Isotherm was created by Irving Langmuir in 1916 (Langmuir, 1916). In its formulation is an empirical model that assumes monolayer adsorption, with adsorption that can only occur at a finite number of definite localized sites that are identical and equivalent, with no lateral interaction and steric hindrance between the adsorbed molecules, even on adjacent sites (Foo and Hameed, 2009). The Langmuir isotherm is defined by Eq. (5), where θ is the fractional surface coverage, K is the Langmuir adsorbate constant, and P_O is the partial pressure of atomic oxygen.

$$\theta = \frac{KP_O}{1+KP_O} \tag{5}$$

Historically, the adsorption model coefficients would have been estimated using laboratory experiments, however, the difficulty with accurately recreating LEO conditions in a laboratory setting can introduce large errors in the models. Contemporary models for satellite applications estimate these empirical values by inverting fitted drag coefficients for satellites with simple shapes (e.g. sphere) (Pilinski et al., 2013; Walker et al., 2014b).

Maxwell's model. Maxwell's model assumes that a fraction ϵ of the particles is specularly reflected, and the remaining fraction $1-\epsilon$ is diffusely reflected. Molecular beam experiments (Hurlbut, 1957; Hurlbut, 1962; Hinchen and Foley, 1966; Kleyn, 2003) have shown that the reflected particles do not follow Maxwell's model and instead follow a quasi-specular model.

Cercignani-Lampis-Lord (CLL) model. Cercignani and Lampis developed the original model in 1971 (Cercignani and Lampis, 1971) by using scattering kernels that describe the relationship between the incident and reflected velocity distribution functions when particles interact with a solid surface. In 1989, R.G. Lord extended the original model to the cases of diffuse reflection with incomplete accommodation and internal energy accommodation (Lord, 1990), thus creating the CLL GSI model. The CLL model ultimately uses the normal energy accommodation coefficient and the tangential momentum coefficient as independent parameters and assumes the quasi-specular deflection kernel.

Diffuse Reflection with Incomplete Accommodation (DRIA) model. In the DRIA GSI model, particles are always reflected with a diffuse angular distribution based on Knudsen's cosine law (Knudsen, 1915). The particles may exchange energy with the surface depending on the value of the energy accommodation coefficient, said to be incomplete accommodation (values between 0 and 1). A special case is when the particles interact with a contaminated surface; it is assumed that the particles adsorb to the layer of atomic oxygen, becoming fully accommodated, and are reflected with a diffuse distribution about the surface normal as illustrated in Fig. 2. (Walker et al., 2014a).

1.2. Total drag coefficient

A quasi-specular drag coefficient model requires relating the effective energy accommodation to the atmospheric properties and such a model was developed by Walker et al. (2014b). Pilinski et al. (2010) showed that the variation of the energy accommodation coefficient assuming diffuse reflection is well-matched by a Langmuir isotherm dependent on the partial pressure of atomic oxygen, P_O. Later work by Pilinski et al. (2013, 2011) was the first to quantitatively test the validity of the Langmuir isotherm model for LEO satellites. The model developed by Walker et al. (2014b) builds on the work of Pilinski et al. (2010) but uses a modified approach with a different scattering kernel and fitting technique. The approach of Walker et al. (2014b) fits a Langmuir isotherm not to the effective energy accommodation coefficient but instead to the fraction of the surface covered by atomic oxygen, θ. Modeling the mixed reflection properties of surfaces with impurities such as adsorbed atomic oxygen through θ was previously suggested by Goodman (1974). The Langmuir isotherm can only model monolayer adsorption; however, P_O at the altitudes of interest for LEO satellites is well below the threshold where only monolayer adsorption can occur. Many other adsorption models exist, but the Langmuir isotherm has been used extensively to model the adsorption of atomic oxygen to satellite surfaces (Pilinski et al., 2010; Moe and Moe, 1967; Hedin et al., 1973). The total drag coefficient can be found using the relation (Walker et al., 2014b):

$$C_D = (1-\theta)C_{D,c} + \theta C_{D,ads} \tag{6}$$

where θ is the fractional surface coverage of adsorbed oxygen, $C_{D,c}$ is the surface drag coefficient based on a clean satellite surface, and $C_{D,ads}$ is the drag coefficient of a surface completely covered by an adsorbate (atomic oxygen). The linear relation between C_D and θ assumes that the adsorbed atomic oxygen uniformly covers the satellite surface.

The $C_{D,ads}$ and $C_{D,c}$ are obtained from the drag coefficients of atmospheric constituent species (H, He, N, N_2, O, O_2) using:

$$C_{D,ads} = \left(\frac{1}{\sum_{k=1}^{6}(\chi_k m_k)}\right)\sum_{k=1}^{6}(\chi_k m_k C_{D,ads_k}) \tag{7}$$

$$C_{D,c} = \left(\frac{1}{\sum_{k=1}^{6}(\chi_k m_k)}\right)\sum_{k=1}^{6}(\chi_k m_k C_{D,c_k}) \tag{8}$$

where χ_k is the mole fraction of species k, m_k is the mass of species k, and $C_{D,ads_k/c_k}$ is the drag coefficient for species k. In our toolkit, the mole fractions of the species are obtained from the NRLMSISE-00 density model (Picone et al., 2002).

For DRIA, following are the independent variables required for the drag coefficient (dependent variable) computation: (i) relative velocity of the satellite, v_∞, (ii) satellite surface temperature, T_w, (iii) atmospheric translational temperature, T_∞, (iv) energy accommodation coefficient, α, (v) satellite yaw, β, and, (vi) satellite pitch, Φ. The C_{D,ads_k} for DRIA is computed using the diffuse reflection with complete accommodation ($\alpha = 1$). The C_{D,c_k} for DRIA is computed based on Goodman's empirical model (Goodman and Wachmann, 1966):

$$\alpha = \frac{K_s \mu}{(1+\mu)^2} \tag{9}$$

where α is the energy accommodation coefficient for the gas particles of the clean satellite surface, and μ is the ratio of the average mass of the atmospheric gases to the mass of the particles that compose the satellite surface. The mass of particles that compose the satellite surface is dependent on the material and finish of the satellite surface. The value of K_s, the substrate coefficient, can lie anywhere from 2.4 to 3.6 depending on the shape of the satellite (Mehta et al., 2017). Mehta et al. (2014) used the coefficient of 3.0 for the original RSM suite which we do not change here. Further analysis of a more appropriate K_s is possible for the user to perform.

For the CLL GSI model, the drag coefficient depends upon the following independent variables: (i) v_∞, (ii) T_w, (iii) T_∞, (iv) normal energy accommodation coefficient, α_n, (v) tangential momentum accommodation coefficient, σ_t, (vi) β, and, (vii) Φ. For CLL, the C_{D,ads_k} is computed using $\alpha_n = 1$ and $\sigma_t = 1$. The clean surface drag coefficient for CLL, C_{D,c_k}, is obtained using $\sigma_t = 1$ and:

$$\alpha_n = \max\left\{2\left(\frac{K_s\mu}{(1+\mu)^2}\right) - 1, 0\right\} \tag{10}$$

where $K_s = 3.0$, and the max function represents the maximum of the two provided arguments.

1.3. Computational analysis methods

1.3.1. Flow regimes

The Knudsen number is used to determine the appropriate computational method for modeling a flow. This nondimensional parameter defines the flow regime in which the analysis is performed. The Knudsen number is given by Eq. (11), where λ is the mean free path and L is the characteristic length of the object.

$$Kn = \frac{\lambda}{L} \tag{11}$$

Fig. 3 shows the classification of flow regimes according to their Knudsen numbers. The FMF regime is recognized as a flow system with a Knudsen number roughly greater than 10, typically assumed to occur at an altitude greater than 150 km (depending on the shape of the satellite. This regime is where the satellites of our interest orbit. This form of flow is thought to occur in high vacuum, where intermolecular collisions are rare, and GSIs are the prevalent phenomena.

Currently, there are several computational analysis methods for modeling satellite drag. For a simple geometric figure like a sphere, we have closed-form solutions for C_D in the FMF regime (Sentman, 1961). However, most satellites do not have simple geometries. The most common computational tools used for computing C_D in the FMF regime are the Test Particle Monte Carlo (TPMC) method and the Direct Simulation Monte Carlo (DSMC) method. The Monte Carlo techniques are popular and are used for various studies to obtain normalized aerodynamic force coefficients. A fundamental difference between the two methods is that the DSMC method simulates particle–particle collisions (although it can be turned off in some DSMC software such as SPARTA (Plimpton et al., 2019)), whereas the TPMC method does not.

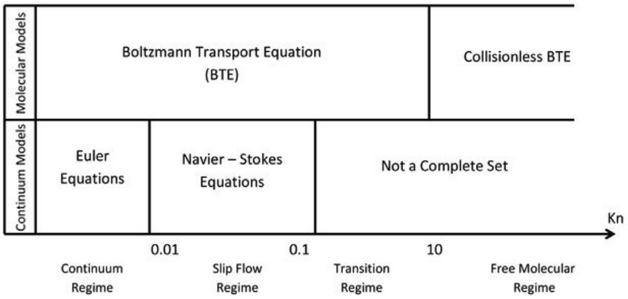


Fig. 3. Classification of flow regimes based on Knudsen number (Bird, 1994).

1.3.2. Direct Simulation Monte Carlo method

DSMC was developed and applied by Graeme Bird in 1963 (Bird, 1963). It is widely used for its high fidelity modeling of flow in transitional flow regimes, where the FMF assumptions no longer hold. The ability to simulate particle–particle collisions is a valuable tool, but it can cause simulations to be computationally expensive.

The satellite geometry is divided into small triangular facets and introduced into the simulation domain. Each facet is an area on the object where properties are defined and calculated. Like most other computational fluid dynamics (CFD) methods, the model fidelity depends upon the quality of the mesh and on how well the surface properties are known.

DSMC operates by inserting molecules into a flow field. To do this, the code uses one of the three methods: it either performs the initial creation of molecules at the first time step, a surface flux from the boundaries, or uses the reservoir method. In the reservoir method, the DSMC simulations are extended to an external region adjacent to the DSMC domain. Particles with the desired velocity distribution are generated in the reservoir at every time step and, those that enter the DSMC domain are accepted while the remaining particles are rejected. In DSMC simulations, like other atmospheric CFD simulations, the computational domain is part of a larger flow environment. Because the computational domain is part of the larger flow environment, the boundary conditions are often set to freestream conditions where molecules are allowed to leave and enter the computational domain, and the number of simulated molecules varies with time. Every simulated particle in the DSMC method represents a number of real gas molecules. This number acts as the statistical weight of a simulated molecule as well.

NASA has created the DSMC Analysis Code (DAC) to study rarefied gas dynamics pertaining to problems such as atmospheric re-entry (LeBeau and Lumpkin, 2001). This code is an export-controlled item that is only accessible to U.S. citizens. DAC can automatically adapt the collision grid to resolve the local mean free path of a flow field. DAC also utilizes a pre-processor that specifies the time step and statistical weight for representative molecules given the gas conditions such as number density and freestream velocity. Furthermore, DAC offers the ability to perform computations in parallel using the Message Passing Interface (MPI). DAC is capable of simulating multiple GSIs including, Maxwell's Model, CLL, and DRIA. These features make DAC a powerful tool. However, it is more computationally expensive than the TPMC method and, therefore, not ideal for the FMF regime where intermolecular collisions are negligible. Unlike the DAC, the earlier mentioned DSMC software SPARTA is open source and has been used by a number of authors including March et al. (2019a), Marín (2019), March (2021), Walsh et al. (2021).

1.3.3. Test Particle Monte Carlo method

TPMC is a Monte Carlo simulation technique first proposed by Davis (1961). TPMC is an excellent modeling tool as it is computationally inexpensive and as accurate as other known models in the free molecular flow regime. A user can model the satellite with pronounced fidelity, often with an error less than 1.0% (Mehta et al., 2014).

TPMC works similarly to DSMC, where each test particle represents a number of real gas molecules. These test particles are sequentially launched into the computational domain. The molecules enter the domain with probabilistically determined velocities. The velocity is composed of a constant freestream bulk velocity and a thermal velocity that is also probabilistically determined. The test particles do not undergo intermolecular collisions. However, TPMC is capable of simulating multiple deflections off the satellite's surface. This applies to complex and concave satellite geometries where flow shadowing is prevalent as well.

1.3.4. Surrogate modeling

The numerical methods used to simulate the drag coefficient are computationally expensive. Even a single simulation instance for a specific input configuration can have a significant computational cost. For orbit propagation of satellites, users may need to perform multiple drag coefficient simulations that correspond to different input parameter values (e.g., different values for atmospheric temperature and composition) at various orbit locations and over time. The solution to this issue is to use an interpolation technique. By running numerical simulations that cover the space of relevant input configurations, it is possible to generate a set of training data that one can use to create an interpolation or surrogate surface. The Gaussian Process Regression (GPR) model is a great tool to accomplish this. GPR is a generic supervised machine learning method designed to solve probabilistic regression problems (Pedregosa et al., 2011).

Los Alamos National Laboratory (LANL) created the so-called "RSM toolkit" that uses the aforementioned technique to calculate drag coefficients (Mehta et al., 2014). The suite is capable of taking user inputs and implementing multiple GSI models, where the user inputs consist of atmospheric properties, satellite characteristics, and GSI parameters. The LANL RSM toolkit uses MPI to parallelize simulations, making it very efficient and ideal for analysis with high-performance computing clusters. With these abilities and resources, the code is capable of analyzing the most complex of satellite geometries in the FMF regime. The code can be found at the following GitHub link: https://github.com/AndrewCWalker/rsm_tool_suite. For further code resources, refer to Appendix B.

2. Methodology

The original RSM toolkit was developed at LANL to provide the community with a resource for simulating and modeling satellite drag coefficients in the FMF regime. The code is used as the basis for the updated RSM software package created at West Virginia University (WVU) with extended capabilities to handle complex satellite geometries that contain "rotating components". Rotating components mean the following: physical components on the satellite that rotate independently of the satellite's main body, e.g., solar panels. Furthermore, the new RSM tool kit is capable of providing uncertainty estimates, which was not possible in the previous version. Known as the WVU RSM, a comparison between the LANL RSM and the new WVU RSM can be seen in Fig. 4.

The WVU RSM software is developed around providing a new user with an easily navigable environment as well as updating the GPR model used in the LANL RSM to a GPR via Python and the Scikit-learn module. This is beneficial as it is more efficient and free of proprietary programming languages. This new suite allows the user to not only execute and automatically develop RSM models but also to execute individual scripts independently, such as: projected area calculations, satellite geometry model rotation and generation, model creation, and model evaluation (C_d computation). Comparing the flowcharts in Fig. 4, there are a couple of notable differences in operation: rotation has been added to the toolkit, and the area script from the LANL RSM has been replaced with a linear n-D interpolation model for projected area computation. The TPMC simulations within the suite use a Cartesian reference frame where the flow is in the positive x-direction, and the z-direction is facing upward, away from the Earth. For satellites with rotating components, the suite operates by taking individual components of the satellite, rotating them, and then combining them to form one object for TPMC simulation. The suite uses JavaScript Object Notation (JSON) files for reading inputs for flexibility of input parameters. This is highly beneficial as a satellite can have any number of components thus creating an n-number of inputs.

The performance of the WVU RSM is seen to be the same as that found by Mehta et al. (2014); however, there is a significant difference in the performance of the regression model creation code. The LANL code took a matter of hours to complete, while the WVU RSM took only a couple of minutes to generate the files for the regression model. Due to multiple variables (e.g., processors, number of simulations, the fidelity of Standard Triangle Language (STL) file representing the satellite geometry) affecting runtime, a full comparative analysis was not performed; however, the difference in run time was quite significant and worth noting. It is also worth noting that there are no limits on the number of simulations or fidelity of the STL file. The user should be mindful of the processing power of their computer, as using an excessive amount of ensemble points

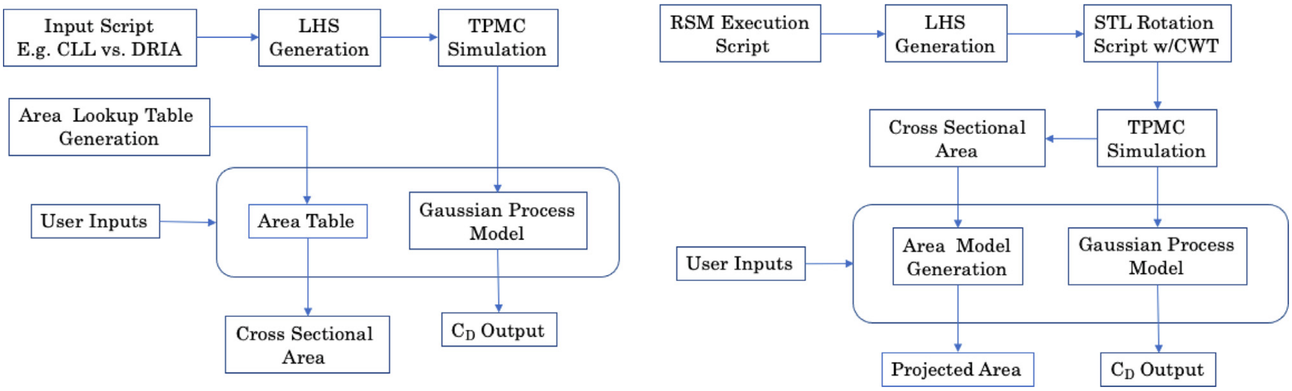


Fig. 4. Flowcharts of LANL (left) and WVU (right) RSM toolkit operation. LHS: Latin hypercube sampling; CWT: “Check Water-Tightness” function.

or triangles in the STL can cause run-time to increase significantly. In general, the satellite or spacecraft geometry needs to have enough triangular facets to accurately represent the shape. For simple convex geometries with curved surfaces such as a sphere, cylinder, or cone, hundreds or even thousands of facets are typically required. For a cube with flat faces, just two triangles on each side are expected to be sufficient. Even for geometries with flat surfaces, meshing requirements become more complex if shadowing effects are significant, whereby more facets may be required to accurately capture the area that is being shadowed. We recommend that for every geometry, the user perform a sensitivity study to arrive at a mesh that is optimized for accuracy and computational cost with TPMC.

2.1. Rotation of STLs

The ability to rotate STL files allows for the creation of a regression model with component rotations as inputs. That is, components of a satellite can be rotated independently of the satellite body and become inputs to the regression model, alleviating the need to create new STL geometries for different configurations of the satellite. A rotation algorithm was used for this software that can rotate individual objects about a user defined axis. This algorithm was extended upon to take the individual rotated STL components and combine them together to form the entire geometry of the satellite. For instance, the CubeSat seen in Fig. 5 is comprised of three parts made by the user using an external modeling tool: two solar panels and a body with connecting arms. Each solar panel is rotated independently by the WVU RSM and then combined with the body to form the satellite geometry for simulation. This rotation code is also its own script within the toolkit and can provide the user with new STL files if they so desire.

To rotate an STL file, an algorithm that was created by James A. Tancred at the Air Force Research Laboratory (AFRL) is used (Tancred, 2012). Originally developed in MATLAB®, the code uses two “hinge point” files to define the line about which the components are rotated. The first

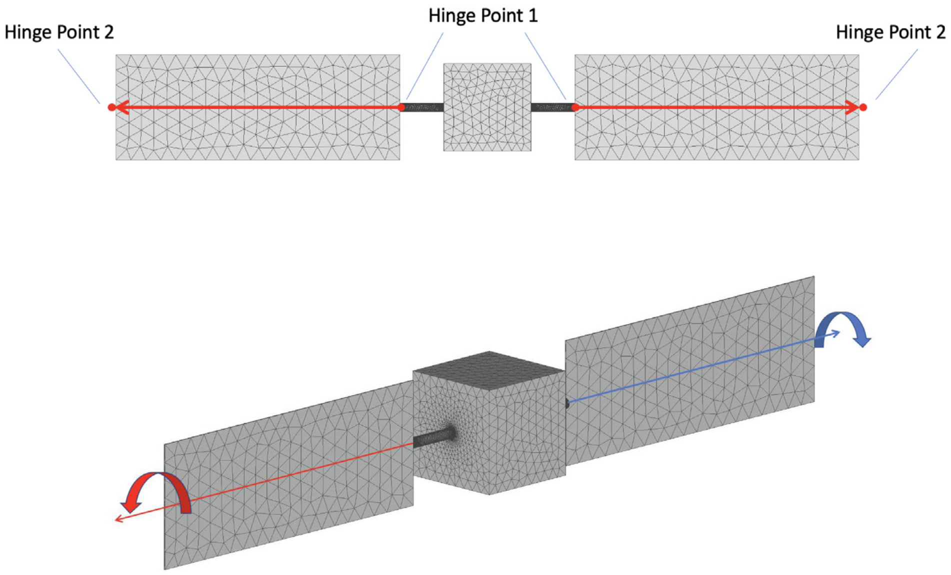


Fig. 5. An example “CubeSat” satellite showing hinge line vector and positive rotation.

file defines the point at which the line of rotation begins, and the second file defines the point at which it ends. These two points define the rotation vector. An example of this can be seen in Fig. 5 where the two hinge lines are defined, and their directions of positive rotation are shown. This algorithm was translated to non-proprietary programming language Python and used to develop the STL models for TPMC simulation.

In order to perform the rotation, each facet needs to be rotated using direction cosine matrices. First, the hinge line vector needs to be defined with respect to the "world origin," a point of reference in the code. The user has the option to define different world origins; however, the natural choice is to use (0,0,0). It should be noted the reference frame used for the rotation is the same as the STL files reference frame. From there, three direction cosines can be calculated by the angles given in Eqs. (12)–(14). These angles are measured by the vector $\overrightarrow{h}$ between the hinge line and the world origin. The cosines of the three rotation angles defining the rotation vector are defined with respect to a Cartesian coordinate frame from the world origin.

$$\alpha = \frac{180}{\pi}\cos^{-1}\left(\frac{h_x}{|\overrightarrow{h}|}\right) \tag{12}$$

$$\beta = \frac{180}{\pi}\cos^{-1}\left(\frac{h_y}{|\overrightarrow{h}|}\right) \tag{13}$$

$$\gamma = \frac{180}{\pi}\cos^{-1}\left(\frac{h_z}{|\overrightarrow{h}|}\right) \tag{14}$$

The rotation angles can be calculated using the following equations:

$$\theta_1 = \frac{180}{\pi}\cos^{-1}\left(\frac{\cos^2(\alpha) + \sin^2(\gamma) - \cos^2(\beta)}{2\cos(\alpha)\sin(\gamma)}\right) \tag{15}$$

$$\theta_2 = 90 - \gamma \tag{16}$$

$$\theta_H = \theta_{deflection} \tag{17}$$

$\theta_{deflection}$ is defined by the users input (e.g.. if the solar panel is to be rotated 45°, then $\theta_{deflection} = 45°$). These three angles can then be used to create the appropriate Eulerian rotation matrices:

$$R_1 = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{18}$$

$$R_2 = \begin{bmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) \\ 0 & 1 & 0 \\ -\sin(\theta_2) & 0 & \cos(\theta_2) \end{bmatrix} \tag{19}$$

$$R_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_H) & \sin(\theta_H) \\ 0 & -\sin(\theta_H) & \cos(\theta_H) \end{bmatrix} \tag{20}$$

The vertices of each facet can now be rotated using the following equation:

$$\mathbf{X} = R_1^{-1}R_2^{-1}R_3R_2R_1(\mathbf{X_b} - \mathbf{\Delta}) + \mathbf{\Delta}; \tag{21}$$

where $\mathbf{X}$ is the new set of vertices for the facet, R_n is the matrix representing the elemental rotations about the respective axes x, y, or z, $\mathbf{X_b}$ is an array of the original vertex points, and $\mathbf{\Delta}$ is the distance between the second hinge point and the world origin. Once the vertices have been rotated, the normal vectors of the facets also need to be rotated. To do this, the same approach and equations are used as the vertices.

After this is accomplished, the new STL components can be combined to create the entirety of the satellite's structure for TPMC simulations. To make sure that there are no errors with this method, a check for water-tightness of the geometries was implemented and is performed on each STL file ensuring functionality with the simulation code. This function makes sure there are no holes in the STL object. Holes could cause particles to become trapped inside the object and give incorrect results.

2.2. WVU regression model

Each input parameter for TPMC simulation is defined over a range. Using this range, Latin hypercube sampling (LHS) provides a set of ensemble designs (input configurations). TPMC produces drag coefficients corresponding to those ensemble designs to create the training data needed. We use GPR, a well-known supervised machine learning method, for fitting a response surface over the training data. An advantage of using the GPR model is that it provides uncertainty estimates.

2.2.1. Gaussian process regression (GPR)

Gaussian process regression is a supervised machine learning technique based on Bayes' theorem. Unlike traditional regression approaches that fit a single function through the observed training data, GPR models the probability distribution over the functional space conditional on the observed data. In GPR, a GP prior is first defined over the functional space $\mathbf{x} \in \mathbb{R}^{D_i} \rightarrow \mathbf{f}(\mathbf{x}) \in \mathbb{R}^{D_o}$. GP is a stochastic process such that for any set of inputs $\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$, the random variables $\{\mathbf{f}(\mathbf{x_1}), \mathbf{f}(\mathbf{x_2}), \ldots, \mathbf{f}(\mathbf{x_n})\}$ are jointly Gaussian. GP prior is defined by a mean function and a covariance matrix:

$$p(\mathbf{F}|\mathbf{X}) = \mathcal{N}(\mathbf{F}|\mathbf{0}, \mathbf{K}) + \mathcal{N}\left(\epsilon|\mathbf{0}, n_l^2\mathbf{I}\right) \tag{22}$$

where $\mathbf{F} = [\mathbf{f}(\mathbf{x_1}), \mathbf{f}(\mathbf{x_2}), \ldots, \mathbf{f}(\mathbf{x_n})]^T, \mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}]^T$, and $\mathcal{N}$ represents the normal distribution. The variable ϵ represents the training data noise with noise-level n_l. The elements of the covariance matrix $\mathbf{K}$ are defined by the "kernel" functions.

The posterior predictive distribution is found using Bayes' theorem by conditioning on the prior and the observations. Given an observation dataset $\{\mathbf{X}_o, \mathbf{F}_o\}$, the predictive posterior distribution corresponding to test inputs $\mathbf{X}_*$ is given by (Rasmussen and Williams, 2006):

$$p(\mathbf{F}_*|\mathbf{X}_*,\mathbf{X}_o,\mathbf{F}_o)=\mathscr{N}(\mathbf{F}_*|\mu_*,\boldsymbol{\Sigma}_*) \tag{23}$$

$$\mu_*=\mathbf{K}_*^T\left(\mathbf{K}_o+n_l^2\mathbf{I}\right)^{-1}\mathbf{F}_o \tag{24}$$

$$\boldsymbol{\Sigma}_*=\mathbf{K}_{**}-\mathbf{K}_*^T\left(\mathbf{K}_o+n_l^2\mathbf{I}\right)^{-1}\mathbf{K}_* \tag{25}$$

with $\mathbf{K}_*=\mathbf{K}(\mathbf{X}_o,\mathbf{X}_*),\mathbf{K}_o=\mathbf{K}(\mathbf{X}_o,\mathbf{X}_o),\mathbf{K}_{**}=\mathbf{K}(\mathbf{X}_*,\mathbf{X}_*)$. Hyperparameters l and n_l are found by maximizing the log marginal likelihood of the training data. The matrix $\mathbf{I}$ represents the identity matrix and the value of $\boldsymbol{\Sigma}_*$ is the measure of the uncertainty of the model.

2.2.2. Implementation

We use the GPR module available with Python's Scikit-learn library (Pedregosa et al., 2011). For the Python GPR implementation, the kernel function needs to be specified by the user. Through much trial and error, we have determined that the most appropriate kernel for our applications is the Matern kernel. The Matern kernel is a stationary kernel and a generalization of the Radial-basis function (RBF) kernel (Pedregosa et al., 2011). The Matern kernel is defined as:

$$k\left(x_i,x_j\right)=\frac{1}{\Gamma(v)2^{v-1}}\left(\frac{\sqrt{2v}}{l}d\left(x_i,x_j\right)\right)^v K_v\left(\frac{\sqrt{2v}}{l}d\left(x_i,x_j\right)\right) \tag{26}$$

where l is a length scale parameter greater than zero, $d\left(x_i,x_j\right)$ is the Euclidean distance, $K_v()$ is a modified Bessel function, and $\Gamma(v)$ is the gamma function. The parameter v controls the smoothness of the function. In this work, v is set to 2.5, a value that is common for twice differentiable functions. As v approaches infinity, the Matern kernel converges to the RBF kernel. When $v=\frac{1}{2}$, the Matern kernel becomes identical to the absolute exponential kernel (Pedregosa et al., 2011).

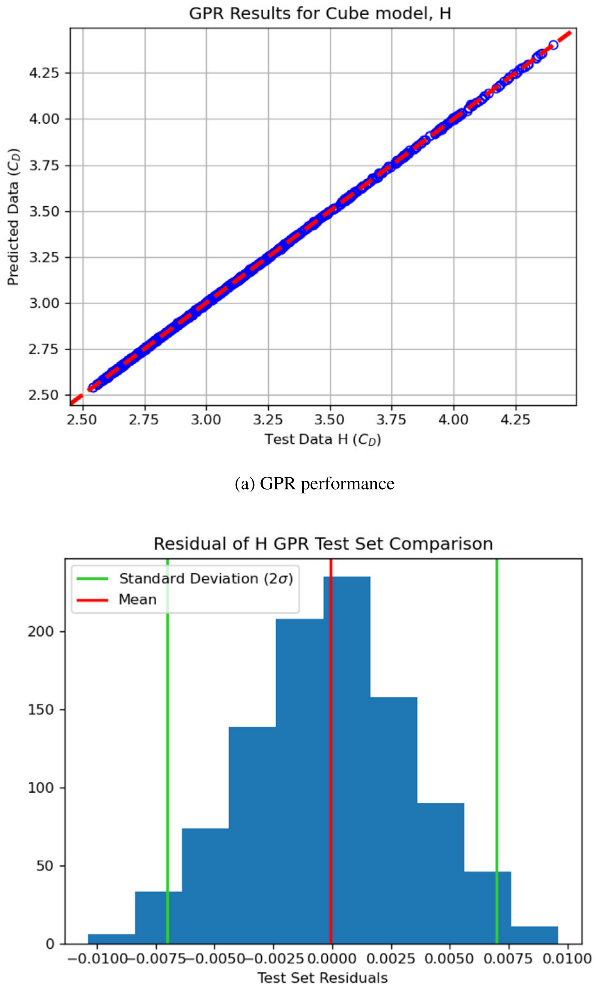
3. Results

3.1. GPR prediction results

If we use too few ensemble points in the training process, i.e., if the ensemble points are sparsely distributed, then the size of the predicted uncertainty is large. If we use too many ensemble points, we have a computationally expensive model at our hand. As such, the configuration and the number of ensemble points need to be carefully designed. For each of the primary constituent species in the LEO region (H,He,N,N_2,O,O_2), 1000 ensemble designs, selected using LHS, are used for training purposes. Another 1000 ensemble designs are used for testing purposes. The simulations performed were done so using the DRIA setting within the software; however, complete accommodation (i.e. $\alpha=1$) was used. Since the purpose of this test is to judge the performance of the GPR, $\alpha=1$ was chosen arbitrarily. Table 1 shows the minimum and maximum values used in the development of the GPR for a cube and a CubeSat. V_∞ is the freestream velocity, T_w is the satellite surface temperature, T_∞ is the atmospheric temperature, β is the yaw, and Φ is the pitch. These ranges were chosen to correspond with values generally seen by satellites in the LEO region. These parameter values were used for both the training and test data for the GPR. Fig. 6(a) shows the comparison between the numerical (TPMC) drag coefficient values and the GPR predictions for a cube for atomic hydrogen (H). Fig. 6(b) shows the histogram of the residuals between the test and predicted drag coefficient values. The predicted values closely follow the numerical test values as evident from the 45° slope of the line in Fig. 6(a). The residual values in Fig. 6(b) are

Table 1
Minimum and maximum for GPR parameters.

Independent Variables	Lower Bound	Upper Bound
V_∞	5500 m/ss	9500 m/s
T_w	250 K	400 K
T_∞	1100 K	2400 K
β	0°	180°
Φ	0°	90°



(a) GPR performance

(b) GPR residuals

Fig. 6. GPR results for a cube for atomic hydrogen.

significantly small numbers, and the histogram is approximately centered at zero. These very low non-biased residuals are an indication of excellent GPR performance. To further quantify the performance of the GPR, the Root Mean Square Error (RMSE) is given for each species in Table 2. The minimum and maximum values for the cube were.00211 and 0.00350, respectively, an indicator that the regression model has high fidelity. The performance corresponding to the other five constituent species is excellent as well. Interested readers should refer to Appendix A, Fig. A.1 for the results corresponding to He, N, N_2, O, O_2 for the cube.

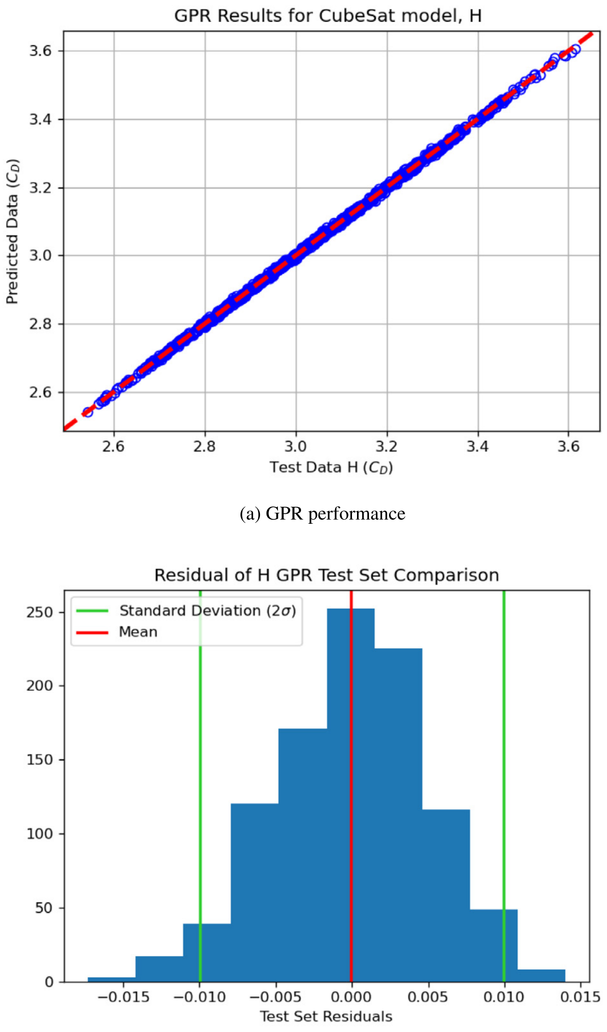
Next, we evaluate a cube with rotating solar panels (CubeSat), as seen in Fig. 5. This satellite consists of a 1mx1mx1m cube body with two solar panels that are each 1mx2m with 10 mm thickness. The rotation of each component now becomes an input to the GPR model creation. In this case, the central body of the satellite does not have any rotation other than pitch and yaw. Therefore, the rotation of the central body is set to zero, giving the GPR an input with zero variance. The code is capable of recognizing the input with zero variance and ignoring it so that it does not have any impact on the GPR's performance. The performance of the GPR for the test data set of atomic hydrogen is shown in Fig. 7. This data is similar to the case of the cube in that the CubeSat also has excellent performance. The RMSE values for the CubeSat are given in Table 2. The CubeSat had a minimum RMSE of.00304 and a maximum RMSE of 0.00498 further proving that the regression model creation is sufficient. For more results, please refer to Appendix A, Fig. A.2.

3.2. Model calibration

Calibration is the requirement in stochastic modeling that the predicted probabilities give an approximation of the likelihood of true events (Camporeale and Caré, 2020). A well-calibrated model, for example, should have around 68% true observations within one standard deviation, 95% true observations within two standard deviations, and 99.7% true observations within three standard deviations. Uncalibrated models tend to be overconfident or under-confident in their predictions, and one should not trust their inferences. A convenient way to check how well a model is calibrated is by looking at its "consistency curve".

Table 2
RMSE for each species for cube and CubeSat satellites.

Species	Cube	CubeSat
H	0.0035	0.00498
He	0.00245	0.00360
N	0.00217	0.00304
N_2	0.00211	0.00311
O	0.00221	0.00311
O_2	0.00213	0.00316



(a) GPR performance

(b) GPR residuals

Fig. 7. GPR results for a cubesat with rotating components for atomic hydrogen.

Let the expected confidence interval levels be: $C = [5\%, 10\%, 15\%, \ldots.., 95\%]$. The corresponding coefficients defining the uncertainty bounds are then given as: $\kappa[k] = \sqrt{2}\ \mathrm{erf}^{-1}(C[k]/100)$, where erf is the well-known error function. Let, $\left(\mathbf{x}_{o_j}, \mathbf{y}_{o_j}\right)_{j=1,\ldots,m}$ be the observation dataset and let the corresponding predictions be $\left(\mu_j, \boldsymbol{\sigma}_j\right)_{j=1,\ldots,m}$, where μ_j represents the mean and σ_j represents the standard deviation. Then, the expected percentage of observed dataset within the lower and upper uncertainty bounds associated with $C[k]$ is obtained as (Anderson et al., 2020):

$$P[k] = \left[\sum_{j=1}^{m} \frac{\mathscr{I}\left(\left(\mu_j - \kappa[k]\sigma_j\right) < y_{o_j} < \left(\mu_j + \kappa[k]\sigma_j\right)\right)}{m}\right] \times 100 \tag{27}$$

where $\mathscr{I}$ is the indicator function.

The consistency curve mentioned earlier is the plot of P versus C. The proximity of the consistency curve to the $y = x$ line (i.e., a straight line with a slope of 45^0 and passing through the origin) is used to measure calibration in this study. The consistency curve will perfectly overlap the $y = x$ line in a perfectly calibrated system. For further explanation of the consistency curve, consider 3σ error bounds. Then ideally, when our model is perfectly calibrated (does not usually happen in real life models), 99.7% of predicted drag coefficients should lie within 3σ standard deviations of the mean (assuming a normal distribution). When we manually count, let's say only 95% of predicted drag coefficients lie within 3σ bounds. The point (99.7, 95) would be one such point in the consistency curve.

For the evaluation of consistency, drag coefficient predictions are made for a sphere and the CHAMP satellite. For each of the objects, 1000 ensemble designs are selected for training, and another 1000 ensemble designs are selected for testing purposes. The training/testing data are created using the numerical TPMC method using the CLL GSI model. The ensemble designs are selected using LHS, where the design space for the LHS sampling is given in Table 3. The consistency plots for sphere and CHAMP drag coefficient predictions using GPR for the test dataset are shown in Fig. 8. The blue dotted line corresponds to the ideal case of perfect calibration, and the green curve corresponds to the GPR predictions. The predictions for the sphere are better calibrated than the predictions for the CHAMP satellite as CHAMP has a more complex geometry than the sphere. For improvements in the model calibration of the CHAMP satellite, we can use more LHS samples, but that would increase the computational cost.

In addition to the qualitative performance of the estimated uncertainties provided in Fig. 8, we also investigate the quantitative performance using the mean absolute calibration error (MACE), defined as:

$$MACE = \frac{1}{n_C}\sum_{k=1}^{n_C}|C[k] - P[k]| \tag{28}$$

where n_C is the number of confidence interval levels.

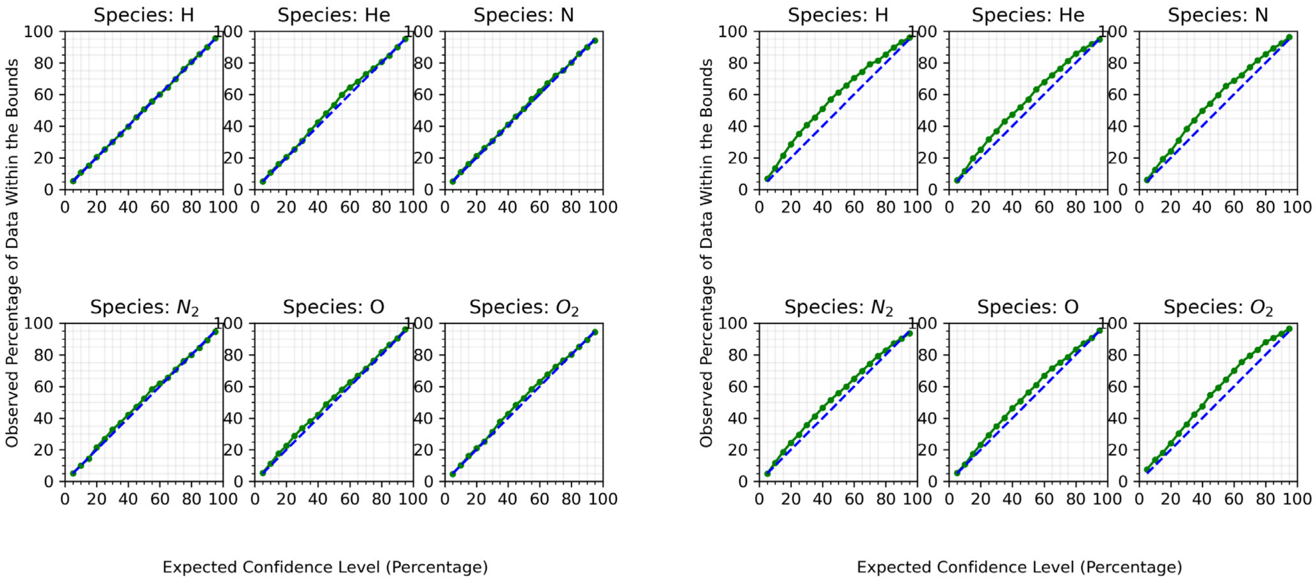
Table 4 lists the MACE values for the sphere and the CHAMP satellites for different atmospheric species. For the sphere, the best MACE value is 0.43%, which corresponds to atomic hydrogen, and the worst MACE value

Table 3
Bounds defining the LHS design points.

Independent Variables	Lower Bound	Upper Bound
V_∞	7250.0 m/s	8000.0 m/s
T_w	100.0 K	2000.0 K
T_∞	200.0 K	2000.0 K
α_n	0.0	1.0
σ_t	0.0	1.0
β	$-10°$	$10°$
ϕ	$-10°$	$10°$

Table 4
Mean absolute calibration error for each species for the sphere and the CHAMP satellite.

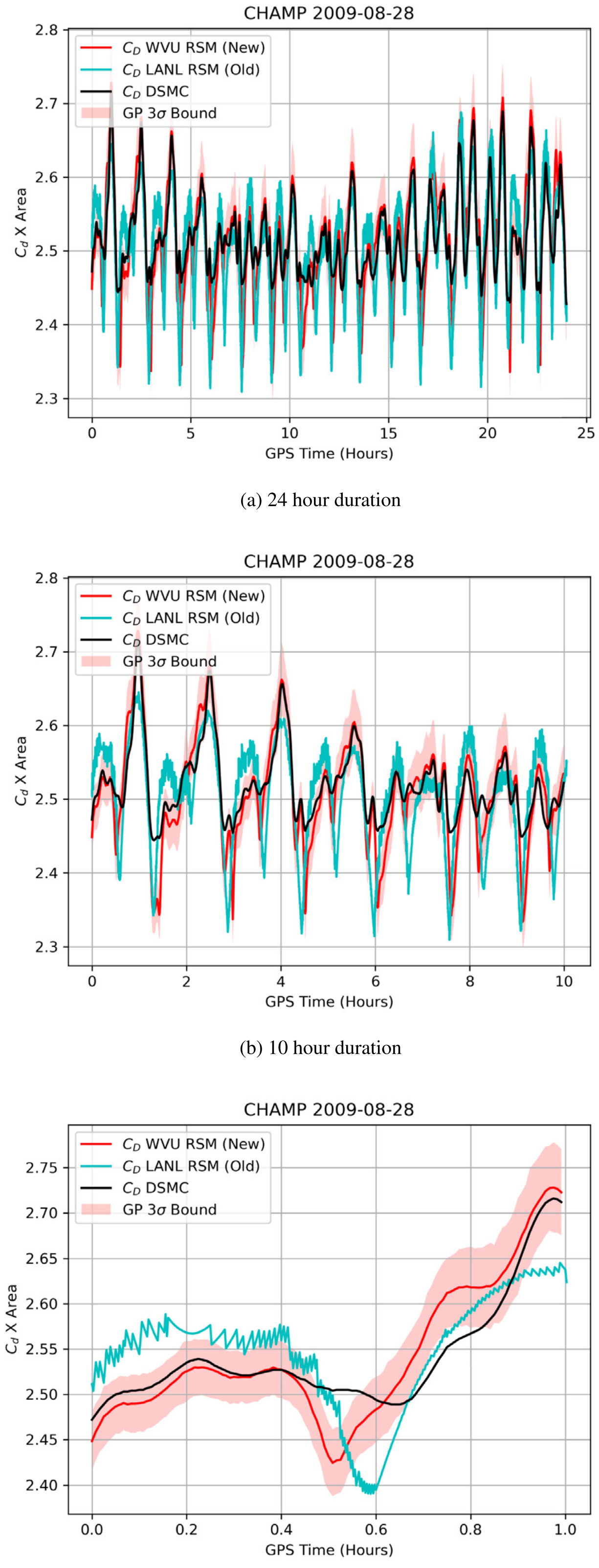
Species	MACE Sphere (%)	MACE CHAMP (%)
H	0.43	7.66
He	1.69	5.37
N	1.03	6.16
N_2	1.34	3.97
O	2.15	3.94
O_2	1.56	6.55



(a) Sphere consistency curve

(b) CHAMP consistency curve

Fig. 8. Consistency plots for a sphere and the CHAMP satellite for drag coefficient prediction using GPR for the test dataset.

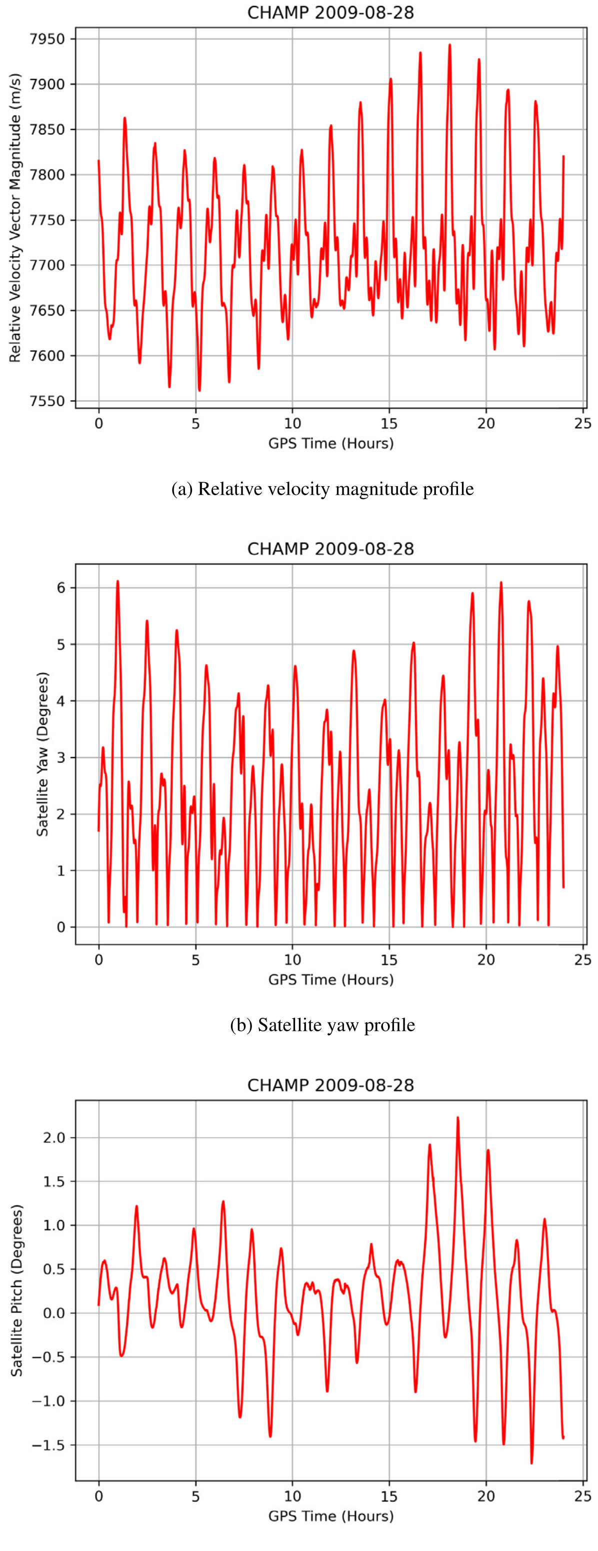


(a) 24 hour duration

(b) 10 hour duration

(c) 1 hour duration

Fig. 9. CHAMP $C_D \times Area$ for August 28, 2009. Comparison of WVU GPR drag coefficient values with other models (in red, we have the WVU GPR model results; in cyan, we have the LANL GPR model results; in black, we have the DSMC results).



(a) Relative velocity magnitude profile

(b) Satellite yaw profile

(c) Satellite pitch profile

Fig. 10. Relative velocity magnitude and satellite attitude profiles for the CHAMP satellite for August 28, 2009.

is 2.15%, which corresponds to atomic oxygen. For the complex CHAMP satellite, the best and the worst MACE values are 3.94% (atomic oxygen) and 7.66% (atomic hydrogen), respectively. These values demonstrate that our models produce meaningful uncertainty estimates.

3.2.1. Computational analysis comparison

To further test the credibility of the WVU RSM Toolkit, a comparison between different models for drag coefficient computation is performed. Fig. 9 compares the LANL and WVU codes that use GPR with the SPARTA-based drag coefficient data provided by Dr. Christian Siemes from the Delft University of Technology (March, 2021). Each of these plots shows the drag coefficient multiplied by the projected area of the CHAMP satellite. The first sub-figure covers one day, while the other two sub-figures are zoomed-in versions with a ten hour duration and one hour duration, respectively. The red-colored curve is calculated using the WVU RSM software for the CLL GSI model, and the cyan-colored curve is generated with the help of LANL RSM software for the CLL GSI model. The satellite surface temperature, which is one of the inputs for the CLL RSM models (refer: Section 1.2), is taken as 400 K. The mass of the particles that compose the CHAMP satellite surface is assumed to be 28 amu. The profiles of the inputs: relative velocity magnitude, satellite pitch, and satellite yaw are given in Fig. 10. The drag coefficient data from the Delft University of Technology (shown in black in Fig. 9) corresponds to the DRIA GSI model, and is generated from DSMC lookup tables. The lookup tables from TU Delft can be downloaded here: http://thermosphere.tudelft.nl. The TU Delft lookup tables (generated for a sequence of accommodation coefficient values) consist of three lookup columns: speed ratio, pitch, and yaw angles; linear interpolation is used for extracting drag coefficient values. The projected area is a function of yaw and pitch and is found by interpolation of an area lookup table generated via each software. Each of these techniques generate projected area using the same CHAMP model. The LANL RSM results are quite noisy since the LANL RSM suite uses bi-linear interpolation on the area lookup table. The WVU RSM model, on the other hand, uses a higher-order interpolation model for the area lookup. The results of the three methods are similar but not the same. The DSMC results, in particular, do not have troughs as low as the RSM results. This probably arises because of differences in TPMC and DSMC simulations as well as the calculation of projected area of the satellite. The LANL GPR results do show a significant amount of noise, but for the most part, lie in close proximity to the 3σ uncertainty bounds of the WVU GPR values. The WVU GPR results follow the DSMC trend quite closely. The RMSE between the mean WVU GPR predictions and the DSMC results is a mere 0.0345. This shows that the WVU GPR provides reasonable results.

4. Conclusions and future possibilities

The goal of the WVU RSM is to provide the community with an efficient tool to estimate drag coefficient, while providing uncertainty, for satellites with and without rotating components. The WVU RSM showed substantial success in generating new STL files, combing STL files to form whole objects, and automatically ensuring that the objects were suitable for simulation (i.e., watertight). The GPR fit by the toolkit shows promise of effectiveness as the residuals/RMSEs between numerically computed drag coefficients and surrogate-predicted drag coefficients were small. The drag coefficient uncertainty estimates were also reliable as illustrated in the consistency or calibration curves. Furthermore, the comparison of CHAMP amongst the different drag modeling software confirms that the WVU RSM performs similarly; however, there is an opportunity for improvement as the LANL RSM and DSMC deviate due to different means of calculating projected area and simulation techniques.

The paper remains incomplete without discussing the potential limitations of the developed toolkit. Since there is a knowledge gap between the operational and the TPMC simulated gas-surface interactions (GSIs) models/model parameters, the fitted response surface is only as good as the assumptions made in the numerical simulations. For example, the work by Bernstein et al. (2020) indicates the possibility of substantial inaccuracies owing to drag coefficient modeling assumptions, particularly at high LEO altitudes. In the future, we may investigate the use of orbit data (Crisp et al., 2021) or estimations of semi-empirical parameters (Ray et al., 2021) to improve the GSI parameters. Furthermore, we acknowledge that the total drag coefficient computation depends upon the species' mole fractions, which are functions of the atmospheric density model. We may explore the effect of different atmospheric density models in the future.

Because we are dealing with regression modeling, there is a prime potential for the use of neural networks instead of the Gaussian process in the WVU RSM toolkit. GPRs typically have a functionality form that is limited to the kernel they use. A neural network could allow for more flexibility in the analysis of data. The neural network also has the capability to handle larger data sets with more efficiency. While the GPR is still efficient, a neural network could extend capabilities to analyze more real-world data.

Furthermore, the WVU RSM is still a software that requires high-level knowledge of command scripting. With the significant increase in efficiency from the Python modeling, the WVU RSM could potentially: (a) move most of the processing to GPUs or a smart combination of CPU and GPU processing, (b) have a web-based graphical user interface (GUI), and the execution happens on a high performance computing (HPC) cluster (or cloud servers) without the user having to know anything about the command-line interface or job submissions, and controlling the input and the output from a web browser, (c) improving the code to use multiple levels of concurrency: multiple job execution, distributed parallelism, multicore parallelism and GPU acceleration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was made possible by NASA West Virginia Space Grant Consortium, NASA Agreement #80NSSC20M0055. The authors would like to acknowledge the use of the High Performance Computing Resources Thorny Flat at West Virginia University housed at the Pittsburgh Supercomputing Center, funded by the National Science Foundation (NSF) Major Research Instrumentation Program (MRI) Award #1726534.

Appendix A. GPR results

See Figs. A.1 and A.2.

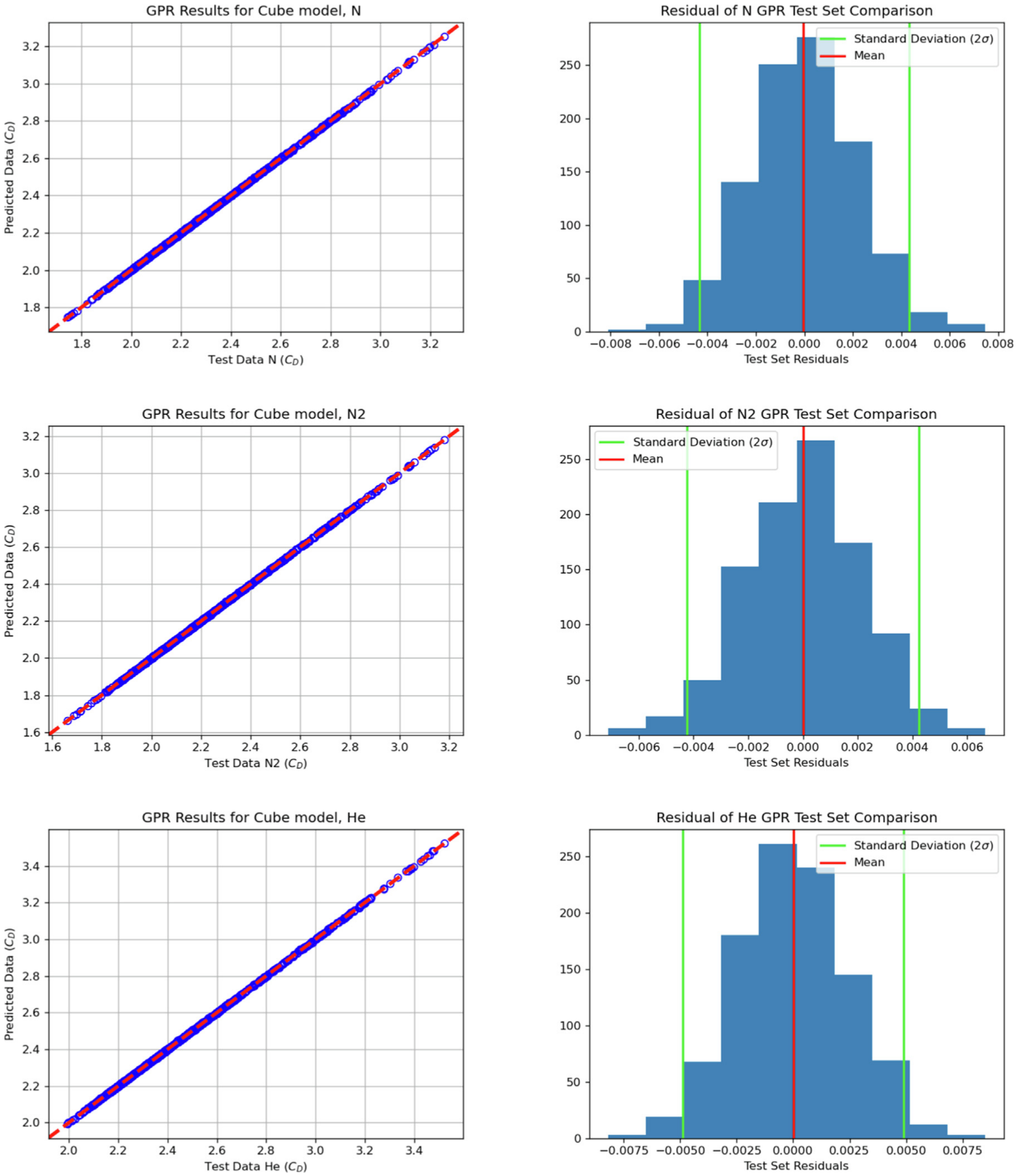


Fig. A.1. GPR results for a simple 1mx1mx1m cube.

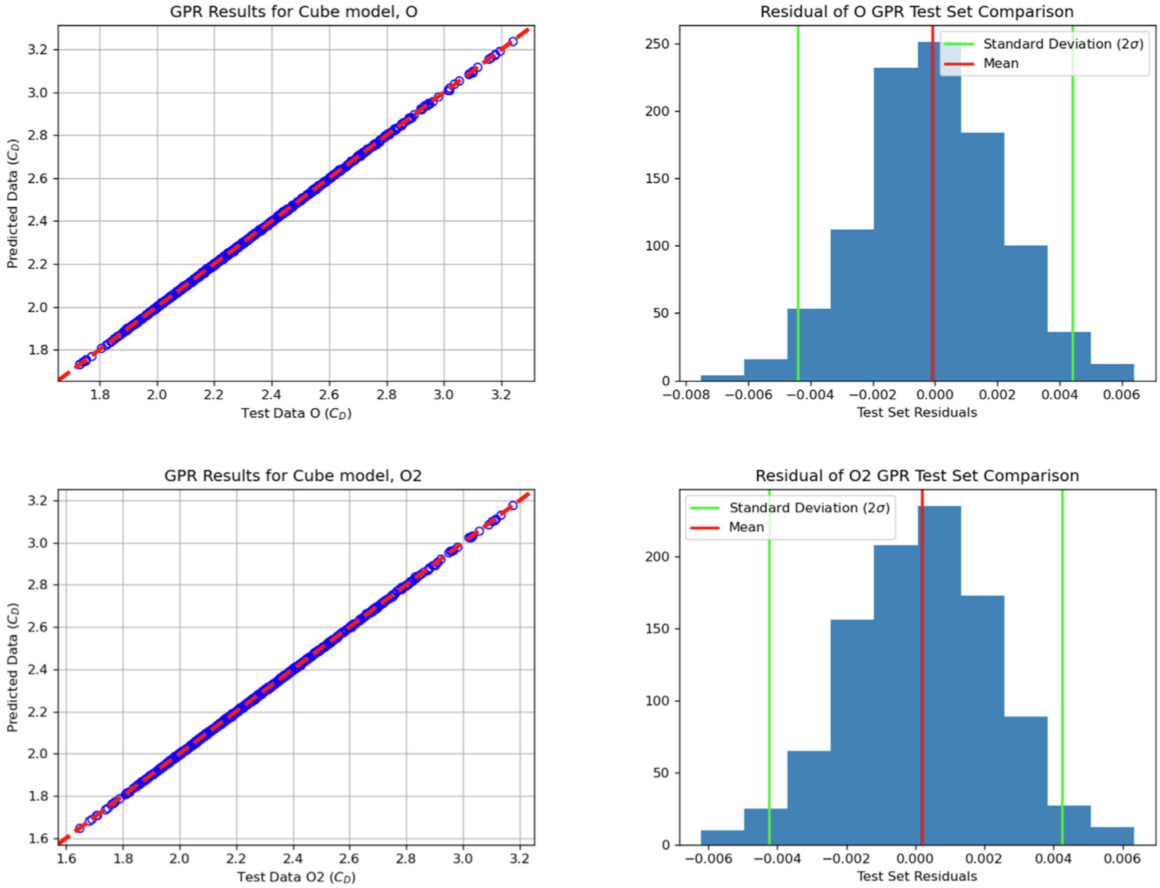


Fig A.1. *(continued)*

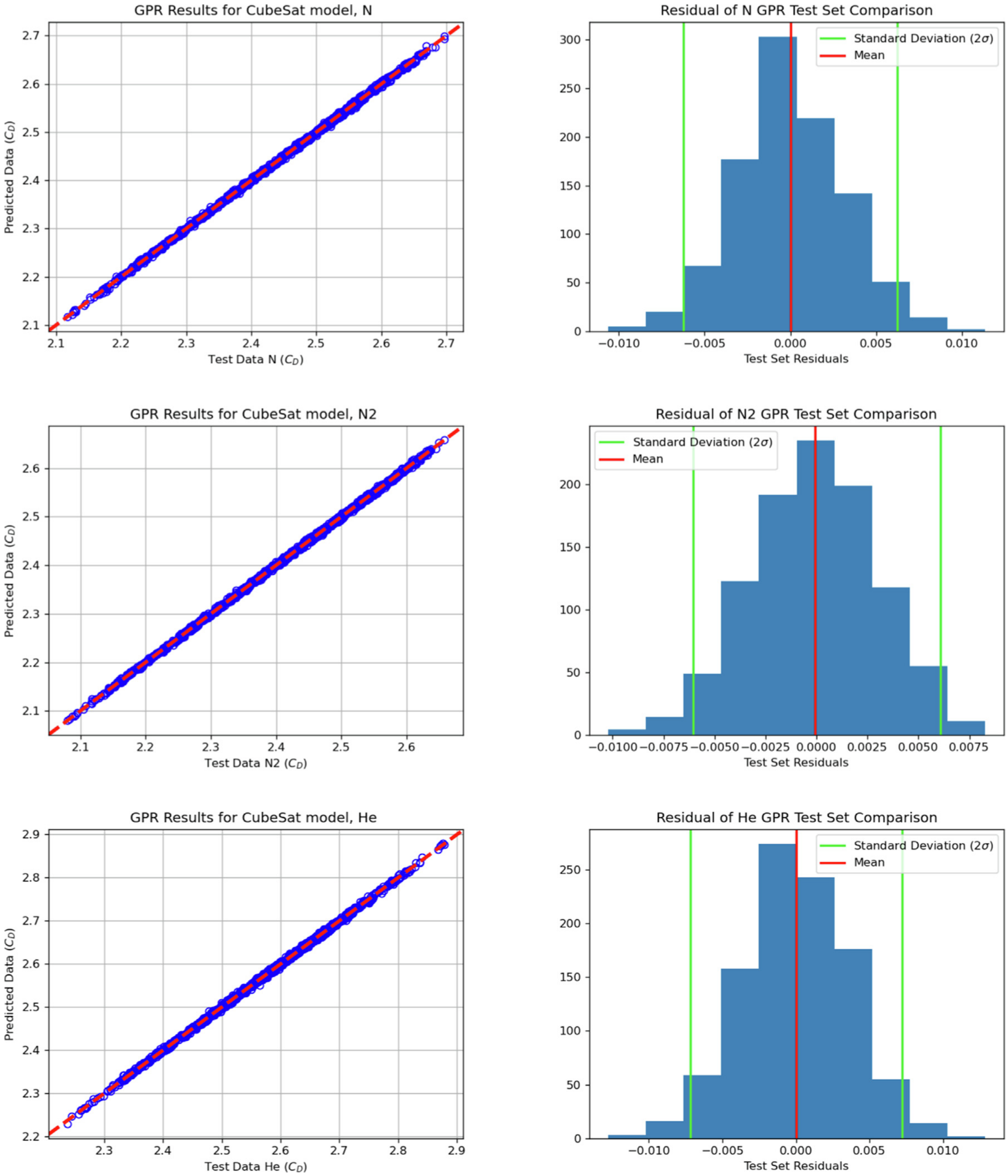


Fig. A.2. GPR results for a CubeSat with two solar panels.

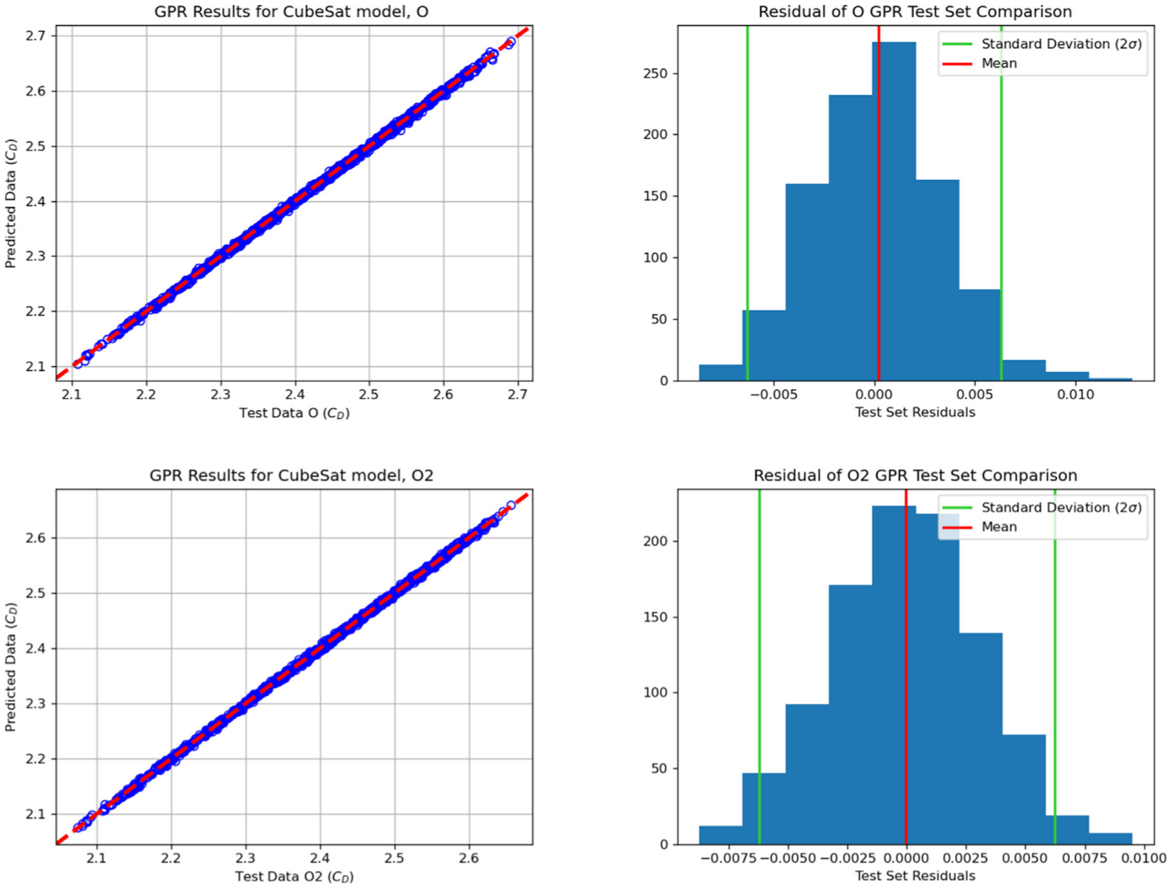


Fig A.2. *(continued)*

Appendix B. Code resources

B.1. Code

The Los Alamos National Laboratory Response Surface Modeling code can be found at: https://github.com/AndrewCWalker/rsm_tool_suite

Reference for this code can be found under the following paper:

Mehta, P. M., Walker, A., Lawrence, E., Linares, R., Higdon, D., and Koller, J. (2014). *Modeling Satellite Drag Coefficients with Response Surfaces*. Advances in Space Research, 54(8), 1590–1607. doi:10.1016/j.asr.2014.06.033

The West Virginia University Response Surface Modeling code can be found at: https://github.com/ASSISTLaboratory/WVU_RSM_Suite

B.2. Developers

Phillip “Logan” Sheridan, Lead Developer, Author
pls0013@mix.wvu.edu

Dr. Guillermo Avendano-Franco, Cluster Specialist, AutoTools.
guavendanofranco@mail.wvu.edu

Dr. Smriti Nandan Paul, GPR Creation.
smritinandan.paul@mail.wvu.edu

Dr. Piyush M. Mehta, Advisor, WVU Assistant Professor.
piyush.mehta@mail.wvu.edu

References

Anderson, G.J., Gaffney, J.A., Spears, B.K., Bremer, P.-T., Anirudh, R., Thiagarajan, J.J., 2020. Meaningful uncertainties from deep neural network surrogates of large-scale numerical simulations. arXiv:2010.13749.

Bernstein, V., Pilinski, M., Knipp, D., 2020. Evidence for drag coefficient modeling errors near and above the oxygen-to-helium transition. J. Spacecraft Rock., 1–18 https://doi.org/10.2514/1.A34740.

Bird, G.A., 1963. Approach to translational equilibrium in a rigid sphere gas. Phys. Fluids 6, 1518.

Bird, G.A., 1994. Molecular Gas Dynamics and the Direct Simulation of Gas Flows.

Camporeale, E., Caré, A., 2020. Estimation of accurate and calibrated uncertainties in deterministic models. arXiv:2003.05103.

Cercignani, C., Lampis, M., 1971. Kinetic models for gas–surface interactions. Transport Theory Stat. Phys. 1, 101–114.

Chambre, P.A., Schaaf, S.A., 2017. Flow of Rarefied Gases. Princeton University Press. https://doi.org/10.1515/9781400885800.

CNBC, 2020. Why in the next decade companies will launch thousands more satellites than in all of history,. URL: https://www.cnbc.com/2019/12/14/spacex-oneweb-and-amazon-to-launch-thousands-more-satellites-in-2020s.html. Accessed 2021-08-01.

Crisp, N., Roberts, P., Livadiotti, S., Macario Rojas, A., Oiko, V., Edmondson, S., Haigh, S., Holmes, B., Sinpetru, L., Smith, K., Becedas, J., Domínguez, R., Sulliotti-Linner, V., Christensen, S., Nielsen, J., Bisgaard, M., Chan, Y.-A., Fasoulas, S., Herdrich, G., Romano, F., Traub, C., García-Almiñana, D., Rodríguez-Donaire, S., Sureda, M., Kataria, D., Belkouchi, B., Conte, A., Seminari, S., Villain, R., 2021. In-orbit aerodynamic coefficient measurements using SOAR (satellite for orbital aerodynamics research). Acta Astronaut. 180, 85–99.

Davis, D., 1961. Monte Carlo calculation of molecular flow rates through a cylindrical elbow and pipes of other shapes. J. Appl. Phys. 31, 1169–1176. https://doi.org/10.1063/1.1735797.

Emmert, J., 2015. Thermospheric mass density: A review. Adv. Space Res. 56, 773–824. https://doi.org/10.1016/j.asr.2015.05.038.

Foo, K.Y., Hameed, B.H., 2009. Insights into the modeling of adsorption isotherm systems. Chem. Eng. J. 156, 2–10. https://doi.org/10.1016/j.cej.2009.09.013.

Freundlich, H., 1907. Über die adsorption in lösungen (in German). Zeitschrift Für Physikalische Chemie 57U, 385–470. https://doi.org/10.1515/zpch-1907-5723.

Goodman, F.O., 1974. Thermal accommodation. Prog. Surf. Sci. 5 (3), 261–375. https://doi.org/10.1016/0079-6816(74)90005-7.

Goodman, F.O., Wachmann, H.Y., 1966. Formula for thermal accommodation coefficient. Massachusetts Inst. of Technology, Fluid Dynamics Lab. Rept., 66-1.

Hedin, A.E., Hinton, B.B., Schmitt, G.A., 1973. Role of gas–surface interactions in the reduction of Ogo 6 neutral particle mass spectrometer data. J. Geophys. Res. 78 (22), 4651–4668. https://doi.org/10.1029/JA078i022p04651.

Hinchen, J.J., Foley, W.M., 1966. Scattering of molecular beams by metallic surfaces. In: Proceedings of the 4th International Symposium on Rarefied Gas Dynamics, Academic Press, New York, pp. 505–512.

Hurlbut, F., 1957. Studies of molecular scattering at the solid surface. J. Appl. Phys. 28 (8), 844–850. https://doi.org/10.1063/1.1722872.

Hurlbut, F., 1962. On the molecular interactions between gases and solids. Univ. of California TR-HE-150-208, Berkeley, CA.

Kessler, D.J., Johnson, N.L., Liou, J., Matney, M., 2010. The Kessler syndrome: Implications to future space operations. In: Proc. 33rd Annual AAS Guidance and Control Conf..

Kleyn, A., 2003. Molecular beam scattering at metal surfaces, surface dynamics. Elsevier, New York, pp. 79–108.

Knudsen, M., 1915. Das Cosinusgesetz in der kinetischen Gastheorie. Ann. Phys. 353 (24), 1113–1121. https://doi.org/10.1002/andp.19163532409.

Langmuir, I., 1916. The constitution and fundamental properties of solids and liquids. Part I. Solids. J. Am. Chem. Soc. 38 (11), 2221–2295.

LeBeau, G., Lumpkin III, F., 2001. Application highlights of the DSMC Analysis Code (DAC) software for simulating rarefied flows. Comput. Methods Appl. Mech. Eng. 191, 595–609. https://doi.org/10.1016/s0045-7825(01)00304-8.

Lord, R., 1990. Application of the Cercignani–Lampis scattering kernel to direct simulation Monte Carlo calculations. In: Proceedings of the 17th International Symposium on Rarefied Gas Dynamics, pp. 1427–1433.

Lord, R.G., 1991. Some extensions to the Cercignani-Lampis gas-surface scattering kernel. Phys. Fluids 3 (4), 706–710. https://doi.org/10.1063/1.858076.

March, G., Doornbos, E., Visser, P., 2019a. High-fidelity geometry models for improving the consistency of CHAMP, GRACE, GOCE and Swarm thermospheric density data sets. Adv. Space Res. 63, 213–218. https://doi.org/10.1016/j.asr.2018.07.009.

March, G., IJssel, J.V.D., Siemes, C., Visser, P.N.A.M., Doornbos, E.N., Pilinski, M., 2021. Gas-surface interactions modelling influence on satellite aerodynamics and thermosphere mass density. J. Space Weather Space Climate 11. https://doi.org/10.1051/swsc/2021035.

March, G., Visser, T., Visser, P., Doornbos, E., 2019b. CHAMP and GOCE thermospheric wind characterization with improved gas-surface interactions modelling. Adv. Space Res. 64, 1225–1242. https://doi.org/10.1016/j.asr.2019.06.023.

Marín, A., Sebasti ao, I.B., Tamrazian, S., Spencer, D., Alexeenko, A., 2019. DSMC-SPARTA aerodynamic characterization of a deorbiting cubesat. In: AIP Conference Proceedings 2132, 070024.

Maxwell, J.C., 1879. Vii. On stresses in rarified gases arising from inequalities of temperature. Philosoph. Trans. Roy. Soc. London, 231–256. https://doi.org/10.1098/rstl.1879.0067.

Mehta, P.M., Walker, A., Lawrence, E., Linares, R., Higdon, D., Koller, J., 2014. Modeling satellite drag coefficients with response surfaces. Adv. Space Res. 54 (8), 1590–1607. https://doi.org/10.1016/j.asr.2014.06.033.

Mehta, P.M., Walker, A.C., Sutton, E.K., Godinez, H.C., 2017. New density estimates derived using accelerometers on board the CHAMP and GRACE satellites. Space Weather. https://doi.org/10.1002/2016SW001562.

Moe, K., Moe, M.M., 1967. The effect of adsorption on densities measured by orbiting pressure gauges. Planet. Space Sci. 15 (8), 1329–1332.

Moe, K., Moe, M.M., Doornbos, E., 2010. Outstanding issues related to thermospheric measurements and modeling. In: 38th COSPAR Scientific Assembly, pp. 4.

Moe, K., Moe, M.M., Rice, C.J., 2004. Simultaneous analysis of multi-instrument satellite measurements of atmospheric density. J. Spacecraft Rock. 41, 849–853. https://doi.org/10.2514/6.2003-570.

Moe, K., Moe, M.M., Wallace, S.D., 1998. Improved satellite drag coefficient calculations from orbital measurements of energy accommodation. J. Spacecraft Rock. 35, 266–272. https://doi.org/10.2514/2.3350.

Moe, M.M., Moe, K., 1969. The roles of kinetic theory and gas-surface interactions in measurements of upper-atmospheric density. Planet. Space Sci. 17, 917–922. https://doi.org/10.1016/0032-0633(69)90097-X.

NASA, 2021. Space debris and human spacecraft,. URL: https://www.nasa.gov/mission_pages/station/news/orbital_debris.html. Accessed 2021-08-01.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. J. Machine Learn. Res. 12, 2825–2830.

Picone, J.M., Hedin, A.E., Drob, D.P., Aikin, A.C., 2002. NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues. J. Geophys. Res. 107, SIA 15-1–SIA 15-16.

Pilinski, M.D., Argrow, B.M., Palo, S.E., 2010. Semi-empirical model for satellite energy-accommodation coefficients. J. Spacecraft Rock. 47 (6), 951–956. https://doi.org/10.2514/1.49330.

Pilinski, M.D., Argrow, B.M., Palo, S.E., 2011. Drag coefficients of satellites with concave geometries: Comparing models and observations. J. Spacecraft Rock. 48 (2), 312–325. https://doi.org/10.2514/1.50915.

Pilinski, M.D., Argrow, B.M., Palo, S.E., Bowman, B.R., 2013. Semi-empirical satellite accommodation model for spherical and randomly tumbling objects. J. Spacecraft Rock. 50 (3), 556–571. https://doi.org/10.2514/1.A32348.

Plimpton, S., Moore, S., Borner, A., Stagg, A., Koehler, T., Torczynski, J., Gallis, M., 2019. Direct simulation Monte Carlo on petaflop supercomputers and beyond. Phys. Fluids 31, 86–101.

Ramillien, G., Cazenave, A., Brunau, O., 2004. Global time variations of hydrological signals from GRACE satellite gravimetry. Geophys. J. Int. 158, 813–826. https://doi.org/10.1111/j.1365-246X.2004.02328.

Rasmussen, C., Williams, C., 2006. Gaussian Processes for Machine Learning. The MIT Press, Cambridge, Massachusetts.

Ray, V., Scheeres, D.J., Pilinski, M., 2021. Inverting gas-surface interaction parameters from fourier drag-coefficient estimates for a given atmospheric model. Adv. Space Res. 68, 1902–1927.

Reigber, C., Flury, J., Rummel, R., Rothacher, M., Boedecker, G., Schreiber, U., 2006. CHAMP mission 5 years in orbit. In: Observation of the Earth System from Space, pp. 3–15. https://doi.org/10.1007/3-540-29522-4_1.

Sentman, L.H., 1961. Free molecule flow theory and its application to the determination of aerodynamic forces. Technical Report Lockheed Missiles and Space Co. INC. Sunnyvale CA.

SpaceNews, 2020. SpaceX launches fifth batch of Starlink satellites, misses booster landing, URL: https://spacenews.com/spacex-launches-fifth-batch-of-starlink-satellites-misses-booster-landing. Accessed 2021-08-01.

Sutton, E.K., 2009. Normalized force coefficients for satellites with elongated shapes. J. Spacecraft Rock. 46, 112–116. https://doi.org/10.2514/1.40940.

Tancred, J.A., 2012. Rotatestl: A Matlab rotation algorithm for the analysis of computational meshes in stereolithography file format. Air Force Research Laboratory, Aerospace Systems Directorate, Wright-Patterson Air Force Base, Air Force Materiel Command, United States Air Force.

Walker, A., Mehta, P., Koller, J., 2014a. Different implementations of diffuse reflection with incomplete accommodation for drag coefficient modeling. J. Spacecraft Rock. 51 (5), 1522–1532. https://doi.org/10.2514/1.A32668.

Walker, A., Mehta, P., Koller, J., 2014b. Drag coefficient model using the Cercignani–Lampis–Lord gas–surface interaction model. J. Spacecraft Rock. 51 (5), 1544–1563. https://doi.org/10.2514/1.A32677.

Walsh, J., Berthoud, L., Allen, C., 2021. Drag reduction through shape optimisation for satellites in very low Earth orbit. Acta Astronaut. 179, 105–121.