Aerodynamic Coefficient Modelling of Cylindrical Space Debris Analogues During Atmospheric Entry

Nathan Lee Donaldson[(1)]

[(1)]University of Oxford, Southwell Building, Osney Mead, Oxford, UK, OX2 0ES

ABSTRACT

A machine learning model for predicting the drag coefficient, C_D of arbitrary cylindrical space debris items is presented. The model is computed using Gaussian Process Regression (GPR) and is trained using a set of 30,000 aerodynamic simulations performed using *RAC*, a hypersonic panel method code. The model accepts the freestream Mach and Knudsen numbers (Ma and Kn, respectively), the freestream vector pitch angle, α, and the cylinder aspect ration, $L/\varnothing$ as inputs, and returns a single drag coefficient for that condition.

The formulation of the GPR model and the methods utilised by *RAC* are described, followed by an evaluation of the accuracy of GPR model predictions when compared to those calculated using *RAC*. A test set of 1,000 random freestream conditions is input into both the GPR model and *RAC*, and the results compared using various statistical metrics. Excellent correlation is demonstrated between both methods, indicating that the GPR model is well conditioned. An improvement in computational efficiency is also observed, with a decrease in execution time of roughly one order of magnitude compared to *RAC*, and a noticeable decrease in RAM usage.

1 INTRODUCTION

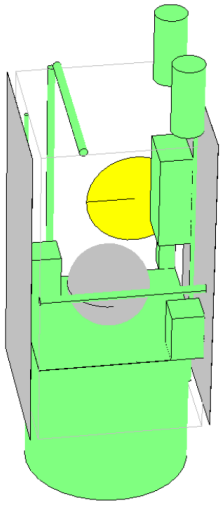
In the study of atmospheric entry, defunct satellites are often modelled using what is known as the "object-oriented" approach (see Figure 1a). This type of analysis involves constructing a numerical analogue of a re-entering spacecraft using geometric primitives to represent its constituent components. Shapes such as cuboids, cylinders, cones, and spheres may be used to represent the likes of electronics enclosures, reaction wheels, fuel tanks, antennae, etc., affording this type of modelling a high degree of versatility.

During a typical analysis of this type, a full model of the re-entering spacecraft is constructed using the geometric primitives described above, and its trajectory propagated through the Earth's atmosphere using 6 DOF (degree of freedom) dynamics. At either a pre-defined altitude or as the result of a transient heat transfer calculation, the breakup of the spacecraft model is simulated. From this point onwards in the simulation, the geometric primitives representing the components are modelled individually, thereby facilitating the calculation of thermal loads and touchdown coordinates (provided of course that they do not burn up during their transit through the atmosphere).

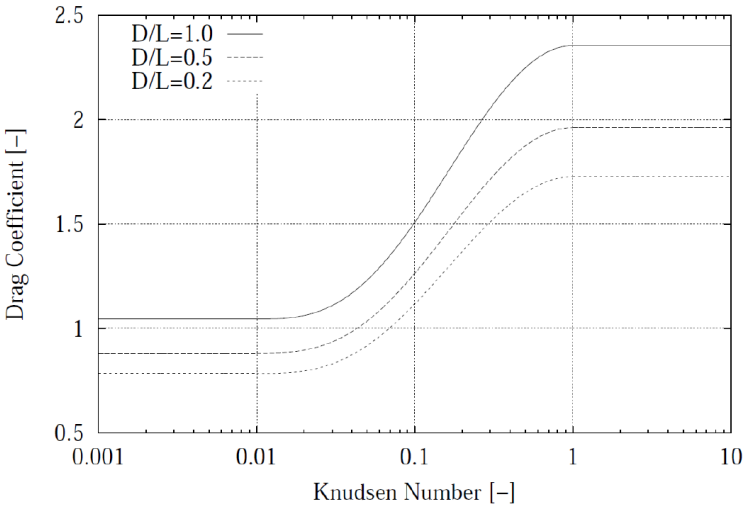
A major drawback of this method as it is currently employed, however, lies in the calculation of the debris analogues' aerodynamic properties. As is mentioned above, the full model of a spacecraft is usually simulated using 6 DOF dynamics, with the relevant aerodynamic derivatives being provided by hypersonic panel methods. Once the breakup event is simulated, however, the individual components are typically assessed aerodynamically using tumble-averaged correlations (see Figure 1b) in order to ensure computational efficiency. Such correlations are a large potential source of uncertainty in debris re-entry calculations, and as such limit the potential accuracy of these otherwise extremely efficient numerical methods. An alternate strategy relies upon a panel method calculation for each component at every step of its trajectory, which can quickly become computationally expensive (especially when performing uncertainty analyses).

In order to improve the accuracy of object-oriented debris re-entry codes, a series of new aerodynamic databases have been generated using a type of machine learning known as GPR (Gaussian Process Regression) modelling. These databases, which are currently implemented for a generic cylinder of aspect ratios in the range $0.1 \leqslant L/\varnothing \leqslant 10$, utilise 4 dimensional covariance functions (kernels) to correlate freestream Mach and Knudsen numbers, the angle of incidence of the cylinder, and the aforementioned aspect ratio. The model is trained on a single aerodynamic coefficient (drag, C_D), with input data being generated using a hypersonic panel method analysis code called RAC (Re-entry Aerothermal Calculator). Once trained, GPR models are extremely computationally efficient, allowing them to feasibly replace tumble-averaged correlations and component-level panel methods in future debris re-entry codes.

Description of the model's construction and the underlying theory of GPR methods are presented, followed by a summary of the simulations performed using RAC. The GPR model for the drag coefficient, C_D, is then tested using a random set of input variables which are also used as input data for further RAC simulations. Next, these two sets of data are compared in order to ascertain the accuracy of the GPR model's predictions compared to RAC. Hence, the ability of such GPR models to potentially improve future debris re-entry calculations is assessed.



(a) Example of object oriented satellite model (NASA-ORSAT) [1]



(b) Kn dependant C_D curves for cylinders of various aspect ratios (ESA-DRAMA) [2]

Figure 1: Examples of current object oriented debris assessment methods

2 GAUSSIAN PROCESS REGRESSION (GPR)

Gaussian process regression (also known as GPR modelling, Kriging, or Wiener-Kolmogarov prediction) is a type of surrogate modelling that has gained a great deal of attention in machine learning research and as an optimisation technique for engineering problems, owing primarily to its generality and scalability. Rasmussen and Williams [3] have exhaustively described the process of formulating and exploiting GPR models as a general machine learning tool, while their use in engineering design was pioneered by Sacks et al. [4], who used them to design and correlate computer simulations. More recently, GPR models have been used by Forrester, Sóbester, and Keane [5] as a major component in engineering optimisation problems. In the field of geostatistics, GPR models are known as Kriging models; named after Danie Krige, a mining engineer who first used the procedure to estimate the density of gold deposits from a small scattering of exploratory boreholes.

Initially, a GPR model is defined solely by its mean and covariance (kernel) functions; this is termed the "prior distribution" (Duvenaud [6] observed that a GPR model's mean is often assumed to be zero, as fluctuations in the magnitude of the mean may be compensated for by adding additional terms to the kernel). Then, as with any surrogate model or regression method, the coordinates of an observed variable ($\mathbf{x}$) are specified, as well as the value of said variable ($\mathbf{y}$) at that point. The GPR model is then "trained" using these data by combining the prior with a Gaussian likelihood function for each of the observed values; this is termed the "posterior". As such, the parameters of the covariance function at each observed point are optimised for the given data set, generating what is known as the "covariance matrix". Once this process is complete, a prediction (y') may be performed by calculating the weighted average of the model at some new point (denoted by the vector $\mathbf{x}'$). This provides the best linear unbiased estimator, as well as the value of the covariance of the GPR model at $\mathbf{x}'$.

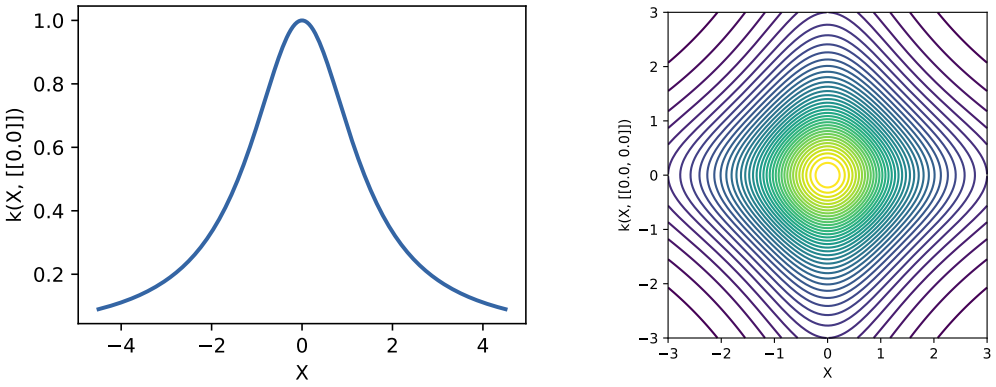
Kernel functions are typically defined by a lengthscale, ℓ, and their variance from the mean, σ_f. Since σ_f is simply a scaling factor, it is the value of ℓ which determines the "relevance" of the function to the value of a predicted variable. As it is unlikely that the response of a system with multiple input parameters will depend on them all to the same degree, a technique known as "automatic relevance determination" (ARD) is used, whereby separate covariance functions are utilised for different input parameters (termed "dimensions" here). These are then added or multiplied

together across the Hilbert space of the GPR model for each observation.

$$K_{RQ,\,ARD}(x, x') = \prod_{d=1}^{D} \sigma_{f_d}^2 \left(1 + \frac{(x_d - x'_d)^2}{2\alpha_d \ell_d^2}\right)^{-\alpha_d} \tag{1}$$

The kernel function, $K_{RQ,\,ARD}$, utilised in the present work is defined as the product of 3 separate rational quadratic (RQ) functions across D dimensions, as shown in Eq. (1). Here, the parameter α allows additional adjustment of the kernel's response (note that, unless otherwise stated, α is used from here on to represent pitch angle). Plots of the kernel function defined in Eq. (1) with $D = 1$ and $D = 2$ are presented in Figures 2a and 2b, respectively.

The software utilised for all of the GPR modelling in the present work is *GPy*, which was developed by the Sheffield Machine Learning Group [7].



(a) $D = 1$ (b) $D = 2$

Figure 2: $K_{RQ,\,ARD}$ kernel functions with one and two input parameters (dimensions)

3 CALCULATION OF HYPERSONIC AERODYNAMICS USING PANEL METHODS

The aerodynamic calculations on which the GPR model is trained were performed using *RAC* [8], a hypersonic panel method code. Panel methods rely upon the inclination of planar surfaces with respect to the velocity vector of the oncoming flow (see Figure 3), and are generally used for the rapid assessment of aerothermodynamics on re-entering objects [9], or for the preliminary design of aircraft [10], missiles [11], launch vehicles [12], and re-entry capsules [13].

These methods were originally developed in the 1960s and, owing to the computational limitations of the era, were used in lieu of the large scale finite volume CFD (computational fluid dynamics) calculations that are widely utilised today.

RAC accepts a triangulated surface mesh of a re-entering object as input, and performs a series of geometric calculations in order to formulate the surface normals and shear vectors. The code also allows the user to set a scale vector whereby the vertices that comprise a mesh may be proportionally shifted along a given axis, thereby altering the aspect ratio of the object. An example of this operation is illustrated in Figure 4. Figure 4b shows the original surface mesh (a unit cylinder), while Figures 4a and 4c show the results of scaling along the longitudinal axis using factors of 0.1 and 10, respectively. Euler angles may also be input by the user in order to facilitate rotation of the freestream vector (as opposed to rotating the mesh itself in order to change the simulated object's attitude).

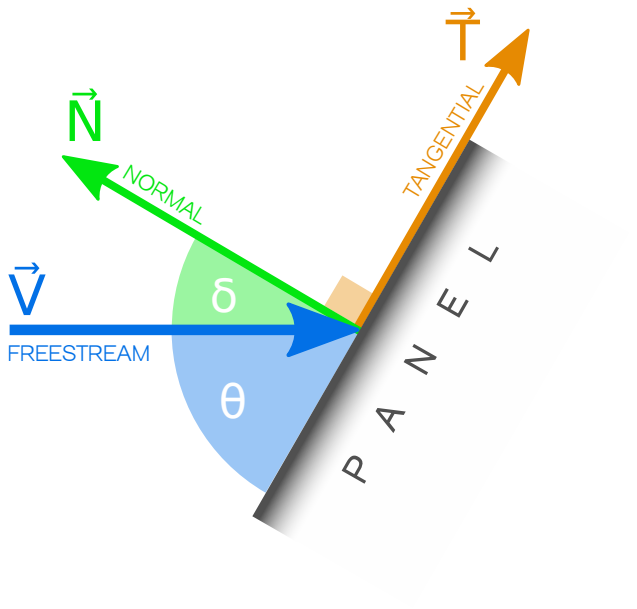
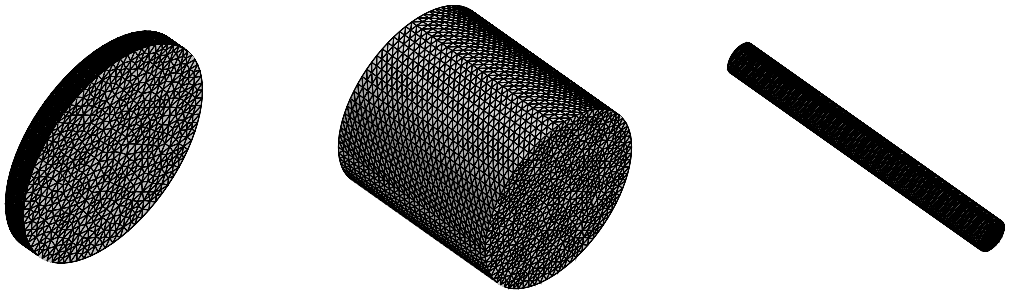


Figure 3: Panel coordinate system and primary vectors of interest for *RAC* calculations

Once the initial processing of the geometry has been completed, *RAC* begins the process of panel shielding calculations. These are designed to identify those panels which would have flow directly incident upon them, and those that would not (and are therefore likely to be in a wake region, or would be obscured by a feature of the geometry). These are referred to as "windward" and "leeward" panels, respectively. In the present work, a basic shielding check was found to suffice, owing to the relative simplicity of the geometry being analysed. This check involved assessing whether the local normal inclination angle, δ, was greater than 90^o on a panel-by-panel basis, thereby identifying which panels were facing the freestream vector (tagged as windward), and those that were not (tagged as leeward).



(a) $\varnothing = 1, L = 0.1$ (b) $\varnothing = 1, L = 1$ (c) $\varnothing = 1, L = 10$

Figure 4: Examples of cylinder surface mesh scaling used in *RAC* analyses

In order to perform its calculations, *RAC* first assesses the freestream Knudsen number for each set of flow conditions input into the solver using Eq. (2). These Knudsen numbers are then grouped into three ranges: continuum ($Kn < 1 \times 10^{-3}$), transition ($1 \times 10^{-3} \leqslant Kn \leqslant 10$), or free molecular ($Kn > 10$). Following these calculations, the appropriate solver for each flow condition is selected.

$$Kn = \frac{\lambda}{L} = \frac{k_B T}{\sqrt{2}\pi\sigma^2 pL} = \frac{Ma}{Re}\sqrt{\frac{\gamma\pi}{2}} \tag{2}$$

Finally, once pressure and shear coefficients (C_p and C_τ) have been obtained using appropriate methods, the freestream conditions are used to calculate the pressure acting on each panel and, in conjunction with the previously

computed normal and shear vectors, the panel force vector. These forces are then summed in each direction of the object's coordinate system and the aerodynamic coefficients calculated. For the present work, the reference area used in these equations was the cross-sectional area of the cylinder, i.e. $A_{ref} = L \cdot \varnothing$.

3.1 CONTINUUM AERODYNAMICS

In the continuum regime, *RAC* may utilise several different methods to calculate the panel pressure coefficients, C_p, although for the present work, only the modified Newtonian method (Eq. (3)) is used for windward panels. The shear coefficient, C_τ is assumed to be zero, owing to the high Reynolds numbers, Re, expected in this regime during re-entry (such high values of Re would indicate a highly inviscid flow where viscous effects are negligible compared to inertial forces).

$$C_{p_{cont}} = C_{p_{0_{cont}}} \sin^2(\theta) \tag{3}$$

Leeward faces in the present work are treated with Eq. (4), which is known as the "70% vacuum" rule [14]. This semi-empirical expression operates on the expectation that recirculation and expansion into the wake region will drive the surface pressure up above the bounding assumption of a full vacuum. As such, it predicts a pressure slightly above vacuum, hence returning a pressure coefficient between Newtonian and full vacuum bounds.

$$C_{p_{cont}} = \frac{-1}{Ma_\infty} \tag{4}$$

3.2 TRANSITION AERODYNAMICS

In the transition regime, which exists between the continuum and free molecular regimes, *RAC* does not directly perform calculations, but rather interpolates from bounding cases.

When a set of flow conditions is preprocessed by *RAC*, its Knudsen number is calculated using Eq. (2). If this value lies in the transition regime as defined above, then *RAC* begins the process of bridging. To perform transition bridging analyses, *RAC* calculates the mean free path lengths, λ, which correspond to Knudsen numbers of 0.001 and 10.0. The program then queries the NRLMSIS00 atmospheric model [15] to find the altitudes at which these values of λ exist, thereby generating two bounding sets of freestream conditions - one at the upper bound of the continuum regime, and one at the lower bound of the free molecular regime.

Simulations are then run on the full range of attitudes using the appropriate continuum and free molecular solvers. As such, pressure and shear coefficient distributions are generated for the boundary cases.

RAC incorporates a global bridging method which utilises the sine-squared function of Wilmoth [16] (Eq. (5)) to bridge aerodynamic coefficients between the continuum and free molecular regimes. The parameters of the function are set as $a_1 = 0.375$ and $a_2 = 0.125$ so that the curve's gradient reduces at the appropriate values of Knudsen number. The force calculation routines must be run on the boundary cases prior to this calculation being invoked.

$$C_{trans} = C_{cont} + \left\{ \left(C_{fm} - C_{cont} \right) \sin^2 \left(\pi \left[a_1 + a_2 \log_{10}(Kn) \right] \right) \right\} \tag{5}$$

3.3 FREE MOLECULAR AERODYNAMICS

In the free molecular regime, *RAC* utilises the analytical methods of Schaaf and Chambre [17] for both C_p (Eq. (6)) and C_τ (Eq. (7)). Due to the nature of the flow in this regime, both windward and leeward faces may be treated effectively with these equations. The hypersonic speed ratio, s, is computed using Eq. (8).

$$\begin{aligned} C_{p_{fm}} = & \frac{1}{s^2} \left\{ \left[\frac{2 - \sigma_N}{\sqrt{2}} s \sin(\theta) + \frac{\sigma_N}{2} \sqrt{\frac{T_w}{T_\infty}} \right] e^{-(s \sin\theta)^2} + \right. \\ & \left. \left[(2 - \sigma_N)((s \sin(\theta))^2 + 0.5) + \frac{\sigma_N}{2} \sqrt{\frac{\pi T_w}{T_\infty}} s \sin(\theta) \right] \left[1 + \operatorname{erf}(s \sin(\theta)) \right] \right\} \end{aligned} \tag{6}$$

$$C_{\tau_{fm}} = \frac{\sigma_T \cos(\theta)}{s\sqrt{\pi}} \left\{ e^{-(s\sin(\theta))^2} + \sqrt{\pi} s \sin(\theta) \Big[1 + \mathrm{erf}(s\sin(\theta)) \Big] \right\} \tag{7}$$

$$s = Ma \sqrt{\frac{\gamma}{2}} \tag{8}$$

3.4 SIMULATION CONDITIONS

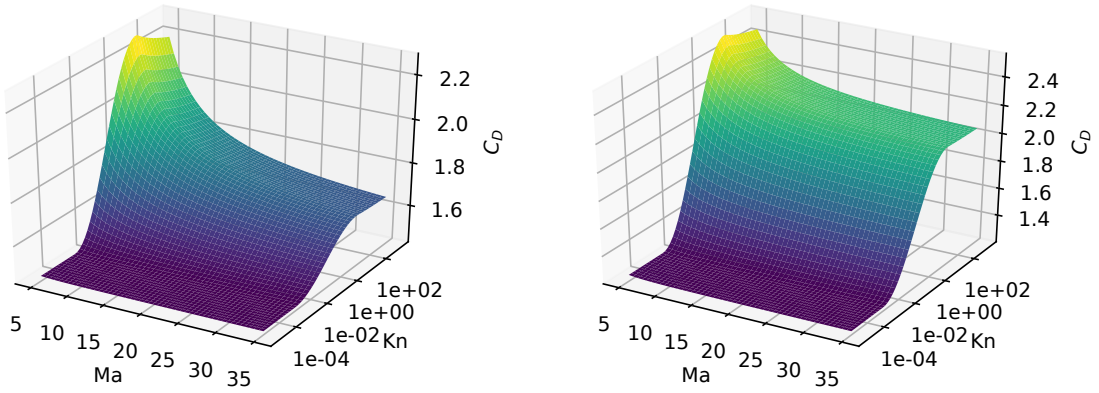
The GPR model was trained with 30,000 *RAC* simulations, the conditions for which were calculated using a Random Latin Hypercube (RLH) sampling plan. The limits applied to this sampling plan are presented in Table 1, whereafter the NRLMSIS00 model was used to calculate the necessary freestream quantities (such as pressure, temperature, velocity, etc.).

Parameter	Symbol	Lower bound	Upper bound	Scale
Mach number	Ma	5	35	Linear
Knudsen number	Kn	1×10^{-5}	1×10^{4}	Logarithmic
Pitch angle	α	0.1^o	89.9^o	Linear
Aspect ratio	$L/\varnothing$	0.1	100	Logarithmic

Table 1: Parameter bounds and scales used in generating input conditions for *RAC* simulations utilising the cylinder geometry.

4 RESULTS

Following training, the GPR model was tested at $\alpha = 0^o$ and $\alpha = 90^o$ for the $L/\varnothing = 1$ cylinder. A range of freestream conditions covering the entire training range of Mach and Knudsen numbers were utilised. The results of these calculations are presented in Figure 5, and show the expected sigmoidal trend of C_D with respect to Kn. A heavy dependence on Ma is evident in the transition and free molecular regimes, with the highest values of C_D occurring as $Ma \to 5$. As expected, the trends approach a true sigmoid as $Ma \to \infty$.



(a) $\alpha = 0^o$, $L/\varnothing = 1$ (b) $\alpha = 90^o$, $L/\varnothing = 1$

Figure 5: Mach-Knudsen surfaces of drag coefficient, C_D, as predicted by the GPR model

In order to quantify the suitability of the GPR model for reproducing aerodynamic data, a random set of 1,000 freestream conditions (Ma, Kn, α, and $L/\varnothing$) were input to both *RAC* and the GPR model (once again using the NRLMSISE00 model to compute additional required quantities). The two sets of results were then compared and errors between them calculated using Eqs (9) through (13).

$$\delta = 100 \left| \frac{y - y'}{y} \right| \tag{9}$$

$$\delta_R = 100 \left| \frac{y - y'}{\max(y) - \min(y)} \right| \tag{10}$$

$$\delta_{RMS} = 100 \sqrt{\frac{\sum_{n=1}^{N} |y_n{}' - y_n|^2}{N}} \tag{11}$$

$$\delta_{NRMS(\mu)} = \frac{\delta_{RMS}}{\overline{y}} \tag{12}$$

$$\delta_{NRMS(R)} = \frac{\delta_{RMS}}{|\max(y) - \min(y)|} \tag{13}$$

The results of these error calculations are presented in Figure 6, and show maximum errors of the order of 1%. The mean percentage error, $\overline{\delta}$, and mean-normalised RMS (root mean square) error were both found to be less than 1%. Normalisation of the errors to the range of the C_D data was also performed (using Eqs. (10) and (13)) in order to provide a more contextualised view of inaccuracies between the GPR model and the *RAC* results. These calculations show that, when considering the full range of returned drag coefficient results, the errors are roughly one order of magnitude smaller than those presented previously, with both the range normalised mean percentage error, $\overline{\delta_R}$, and the range normalised RMS error, $\delta_{NRMS(R)}$, being less than 0.1%.

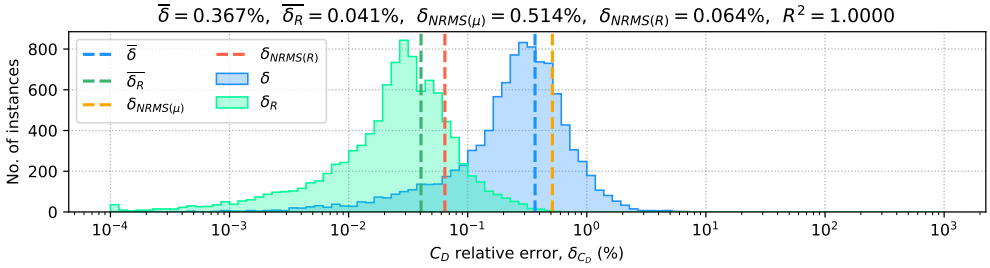


Figure 6: Errors in C_D predictions between *RAC* and the GPR model

Finally, the probability density of the two methods' predictions were computed using the *SciPy* library [18]. The results of these calculations are presented in Figure 7, and show excellent correlation between the GPR and *RAC* results, thereby indicating that the probability of obtaining a given drag coefficient is virtually identical for both methods.

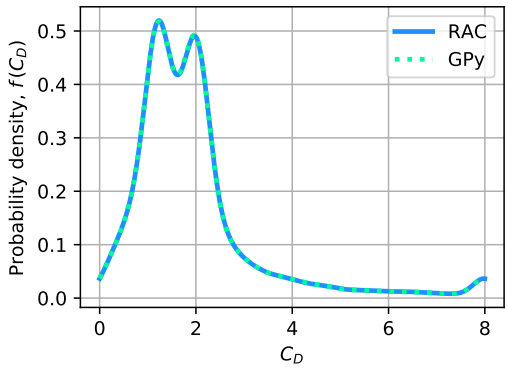


Figure 7: Probability density of C_D predictions generated by *RAC* and the GPR model

5 CONCLUSIONS

In this paper, a machine learning model of the drag coefficients, C_D,for cylindrical space debris has been described. The model, which utilised a correlation method called Gaussian Process Regression (GPR) was trained using 30,000 aerodynamic calculations which were themselves performed using the *RAC* hypersonic panel method code. The GPR model was correlated with fresstream Mach and Knudsen numbers (Ma and Kn, respectively), the freestream vector pitch angle, α, and the cylinder's aspect ratio, $L/\varnothing$.

When queried with a randomised set of input conditions, excellent agreement was found between the model and *RAC*, with peak errors of $\sim 1\%$, and RMS errors of 0.514% (when normalised to the mean C_D result) and 0.064% (when normalised to the full range of C_D results). The probability density of each set of C_D results was then computed, and also showed excellent correlation, thereby supporting the conclusion that the GPR model has captured the underlying trends and magnitudes of the training data set well.

Both *RAC* the GPR model were timed during execution, with the GPR model demonstrating a significant decrease in RAM usage and a decrease in execution time of approximately one order of magnitude when compared to *RAC*. This is significant since panel methods are known to be some of the most computationally efficient calculations available for aerodynamics. The resource usage and execution time of the GPR model is independent of mesh size, and the model itself may be trained using hybrid data sets which include both inexpensive data (such as those generated by *RAC*) and expensive data (such as those derived from experiments or high fidelity simulations).

Future work will include the generation of additional models for other aerodynamic coefficients such as lift, C_L, and pitching moment, C_{my}. Further GPR models will also be generated for different geometric primitives such as cuboids, and will be integrated into an appropriate trajectory solver. This will then be used to test the effectiveness of the GPR models compared to existing tumble-averaged aerodynamic correlations.

Overall, it is clear that the GPR model has effectively reproduced the drag coefficient variation for a cylinder across a wide range of attitudes, Mach numbers, Knudsen numbers and aspect ratios. It is expected that models like this will be incorporated into future debris demise codes in order to both expedite calculations, and increase their accuracy. Such improvements are expected to lead to reductions in cases where debris survive the re-entry process and endanger personnel and property on the ground.

6 REFERENCES

[1] R. L. Kelley, N. M. Hill, W. C. Rochelle, N. L. Johnson, and T. Lips, "COMPARISON of ORSAT and SCARAB Reentry Analysis Tools for a Generic Satellite Test Case," English, Bremen, Germany, Tech. Rep., 2010. [Online]. Available: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20100005304.pdf%20files/13/Kelley%20et%20al.%20-%202010%20-%20COMPARISON%20of%20ORSAT%20and%20SCARAB%20Reentry%20Analysis%20To.pdf.

[2] J. Gelhaus, "DRAMA Software User Manual," ESOC, Darmstadt, Germany, Tech. Rep., 2014.

[3] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006, ISBN: 026218253X. DOI: 10.1142/S0129065704001899. arXiv: 026218253X.

[4] J. Sacks, W. J. Welch, J. S. B. Mitchell, and P. W. Henry, "Design and Analysis of Computer Experiments," *Statistical Science*, vol. 4, no. 4, pp. 409–423, 1989, ISSN: 08834237. DOI: 10.2307/2245858. [Online]. Available: http://dx.doi.org/10.2307/2245858.

[5] A. Forrester, A. Sobester, and A. Keane, *Engineering design via surrogate modelling: a practical guide*. 2008, ISBN: 0470770791. [Online]. Available: http://eprints.soton.ac.uk/64699/.

[6] D. K. Duvenaud, "Automatic Model Construction with Gaussian Processes," no. June, p. 144, 2014. [Online]. Available: https://www.cs.toronto.edu/%7B~%7Dduvenaud/thesis.pdf.

[7] Sheffield Machine Learning Group, *GPy: A gaussian process framework in python*, http://github.com/SheffieldML/GPy, Department of Computer Science, University of Sheffield, 2012.

[8] N. L. Donaldson and P. Ireland, "A Panel Method Aerodynamic Preprocessor for Planetary Entry Trajectory Simulations," in *21st AIAA International Space Planes and Hypersonics Technologies Conference*, 2017, pp. 6–9, ISBN: 978-1-62410-463-3. DOI: 10.2514/6.2017-2379. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2017-2379.

[9] J. Merrifield, J. Beck, G. Markelov, and R. Molina, "Simplified aerothermal models for destructive entry analysis," in *Proceedings of the 8th European Symposium on Aerothermodynamics for Space Vehicles*, Lisbon, Portugal: ESA, 2015.

[10] E. Gentry, Arvel, D. N. Smyth, and W. R. Oliver, *The Mark IV Supersonic-Hypersonic Arbitrary Body Program; Volume II - Program Formulation*, English, 1973.

[11] C. Rosema, J. Doyle, L. Auman, M. Underwood, and W. B. Blake, "AFRL-RB-WP-TR-2011-3071 Missile DATCOM User's Manual - 2011 Revision.," vol. 3071, 2011.

[12] R. Wuilbercq and R. E. Brown, "RAPID AERO-THERMODYNAMIC ANALYSIS FOR HYPERSONIC AIR VEHICLES," in *Proceedings of the 8th European Symposium on Aerothermodynamics for Space Vehicles*, Lisbon, Portugal, 2015.

[13] D. Kinney, "Aerothermal Anchoring of CBAERO Using High Fidelity CFD," pp. 1–28, 2007. [Online]. Available: http://www.aric.or.kr/treatise/journal/content.asp?idx=96691.

[14] D. J. Kinney, J. V. Bowles, L. H. Yang, and C. D. Roberts, "Conceptual Design of a 'SHARP'-CTV," in *Proceedings of the 35th AIAA Thermophysics Conference*, Anaheim, California, USA, 2001. [Online]. Available: https://engineering.purdue.edu/AAE450s/generaldesign/sharp-ctv-01-2887.pdf.

[15] J. Picone, A. Hedin, and D. Drob, "NRLMSISE-00 Empirical model of the atmosphere statistical comparison and scientific issues," *Journal of Geophysical Research*, p. 70, 2001. [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20020038771.pdf.

[16] R. G. Wilmoth, R. C. Blanchard, and J. N. Moss, "Rarefied transitional bridging of blunt body aerodynamics," *21st International Symposium on Rarefied Gas Dynamics*, pp. 26–31, 1998.

[17] S. A. Schaaf and P. L. Chambré, *Flow of rarefied gases*, ser. Princeton aeronautical paperbacks 8. Princeton University Press, 1958.

[18] E. Jones, T. Oliphant, P. Peterson, *et al.*, *SciPy: Open source scientific tools for Python*. 2001. [Online]. Available: http://www.scipy.org/.