Development of Numerical Approaches to Assess Solid Rocket Motor Designs

TYLER WALLENTINE

12 DECEMBER 2023

**ABSTRACT**

When it comes to fuel types for use in various propulsion systems, from rockets and boosters to missiles, solid fuel configurations present a variety of advantages over alternative fuel types in terms of production cost, response time, and overall power. In considering the design of a solid rocket motor, a variety of important dimensional and geometric decisions must be made in the context of the vehicle’s purpose. Total rocket and chamber mass, fuel composition, and fuel geometry are three primary, interlinked variables that play a major role in rocket thrust and trajectory. While numerical analysis of trajectories involving these factors are somewhat straightforward, more diverse and computationally intensive processes are required when it comes to simulating changes in motor geometry. This work outlines numerical frameworks for running image-based simulations of various rocket designs and evaluates their performance using ODEs handling altitude and velocity. Forward Euler and Crank-Nicolson methods are used in tandem to simulate trajectory and are compared. Finally, a comparison between burn simulation frameworks, seeking to classify each approach in terms of computational intensity and outcome accuracy.

**1. INTRODUCTION**

When it comes to various options for fuel types in propulsion systems, from rockets and boosters to missiles, solid fuel configurations present a variety of advantages over alternative fuel types. Solid rocket motors distinctly excel when it comes to long-term storage. Solid propellant facilitates rapid launch and response even after storage, with lifetimes often exceeding 20 or 30 years. Rapid response is a key feature required of modern nuclear systems that seek to respond to nuclear and other threats, including both retaliatory strikes and the interception of incoming attacks (Refuto, 2011).

In their simplest form, solid rockets are necessarily comprised of a casing, igniter, nozzle, insulation, and an internal propellant grain (Figure 1). The working principle supporting flight of the motor is described in a sequence. First, an igniter expels flames down an open cavity in the rocket motor, igniting the innermost exposed layer of propellant grain, beginning the first, unsteady stage of flight. Second, a combustion reaction begins in which the innermost layer of propellant forms a gaseous mixture, creating a pressure gradient causing a gas flow out of the nozzle. Third, choked flow conditions are reached as gas generation from the propellant exceeds the rate at which it can be expelled from the nozzle, increasing the internal pressure of the cavity and producing thrust. The second stage of flight, in which the pressure achieves a semi-steady state, is reached shortly after choked flow begins and chamber pressure increases. The final stage of flight, termination, is reached once the propellant is fully consumed and the rocket chamber pressure equilibrates with the environment (Baars, Tinney, Ruf, Brown, & McDaniels, 2012; Evans, 2010; Tanaka & Shibasawa, 2015).

The rate of gas generation in solid propellant is obviously dependent on the amount of propellant burning at a given time, which is directly related to the surface area exposed to the void region in the center of the rocket. High surface area at the start of flight contributes to a high initial thrust, as well as to either a rapid attainment of slower, steady state burning or flight termination, depending on if the particular propellant geometry reduces to a lower surface area before burning out entirely.

Considering the substantial impact of propellant grain surface area on motor performance, then, it becomes clear that propellant surface area can be modified to achieve particular flight outcomes. However, especially for more complicated geometries, it is difficult to precisely account for changing areas over time manually and seems to necessitate the usage of numerical simulations to compute surface area as a function of A black and white rectangular sign with a number

Description automatically generatedtime, burn rate, and other characteristics worth considering.

**FIGURE 1.** Fundamental design of a solid rocket motor. (1) Motor case, (2) Igniter, (3) Nozzle, (4) Insulation, (5) Solid propellant grain

**2. GOALS & HYPOTHESES**

In essence, the key questions that this work seeks to answer are as follows:

* What is the impact of different grain geometries on the ultimate performance of a rocket, and how can this be modelled?
* What is the “sweet spot” for a rocket-to-fuel mass ratio that allows for sufficient performance while leaving room for non-propulsive mass?
* At what point (if ever), if propellant is scaled up equally with overall rocket mass, does size begin to have an overall detrimental impact on rocket motor performance?

The definition of what makes one rocket’s performance superior over another may vary but, given the fact that trajectories will be the key metric in this project, a “superior” performance will be defined as the highest altitude achieved by a simulated motor.

This work presents some hypotheses in regard to the above questions. These are as follows:

* Surface areas that are reasonably large at the start of flight will allow for altitude maximization by achieving an early, large thrust. This will suggest a limit proportional relationship between propellant surface area and max altitude. A limitation to this relationship will become apparent with extremely multifaceted grain designs, which will expectedly burn too quickly to sustain flight.
* Larger rockets across all grain designs tested will achieve increasing altitudes for propellant compositions greater than or equal to 80%. At lower propellant compositions, benefits from size will rapidly decline due to associated increases in drag with larger frontal surface areas. Thus, in the case of *longer* rockets with no increase in diameter, this decrease is not expected.
* As rocket mass is scaled up, as long as the proportion of fuel relative to overall mass is the same, rocket trajectories will improve and achieve higher maximum altitudes.

A diagram of a circle with a circle and a circle with a circle and a circle with a circle and a circle with a circle and a circle with a circle and a circle with a circle with

Description automatically generated**FIGURE 2.** Axial cross sections of simple geometric grain designs that can be easily estimated analytically. From left to right, these include the tubular, rod and tube, and cross designs.

The first focus of this work will be to set forth different methods for estimating surface area. First, methods including analytical solutions for simple areas will be discussed, as well as their limitations and benefits. A similar description of the various iterations of the numerical methods used to compute area will then follow.

A single one of these methods will be selected to be the primary means behind area calculations during the simulation. Once selected, the relevant ordinary differential equations (ODEs) will be numerically estimated in order to obtain the estimated trajectories for a variety of designs using Forward Euler and Crank-Nicholson methods.

Following completion of the grain-dependent trajectory estimations, additional tests will be run to evaluate the impact of other dimensional variables on trajectory: namely, (1) total motor weight, (2) % starting weight composition of the propellant with respect to overall rocket mass, and (3) rocket motor diameter, playing a role both in the drag and weight the rocket faces.

The final assessment included as part of this work will be an evaluation of the potential for stability analyses. The possibility of performing an eigenvalue analysis will be evaluated—however, the lack of analytical solutions for the majority of motor configurations will likely prevent this.

**3. METHODS**

3.1. SUMMARY OF EQUATIONS

With trajectory being the primary consideration behind the success of motor designs in this work, the following ODEs with respect to time will be used to model flight paths**:**

where and are functions to compute changing gravity and density as a function of altitude . Acceleration is a function of these values in addition to velocity , drag coefficient , frontal surface area (not to be confused with propellant surface area), rocket mass , exit velocity , and the rate of change of fuel mass, The first term of represents gravity, the second term represents drag, and the third term represents thrust.

The Forward Euler method, an explicit method for solving ODEs, will be applied to solve the ODEs presented here, in addition to the implicit Crank-Nicolson methods. The Forward Euler and Crank-Nicolson methods are summarized respectively as follows:

where is treated as either altitude *h*, velocity *V*, or mass *m*, and *f* is the right hand side to the corresponding *y* derivative with respect to time.

The rate of change of fuel mass, , is computed by the following relation to propellant density and surface area in addition to Vielle’s law (DeLuca, et al., 2010; Sutton & Biblarz, 2016):

where is the chamber pressure and where and represent the burn rate coefficient and pressure exponents, respectively (DeLuca, et al., 2010). The process of modeling chamber pressure over time stands to complicate this problem significantly. Accurately accounting for pressure requires one to account not only for contributing chemical reactions and thermodynamics as they pertain to pressure, but also the effects of fluids as turbulent flow, acoustic instabilities, and vortices. Due to the complexity of modelling chamber pressure changes, this model will assume a pressure steady state for the duration of this study.

With the steady state assumption of pressure, propellant surface area plays perhaps the most significant, dynamic role as a variable in computing altitude and velocity. A central question in this study, then, becomes how to best account for propellant surface areas it varies with time.

3.2. ANALYTICALLY ESTIMATING SURFACE AREA

This section describes the approach taken to model surface area for a variety of shapes, as well as its benefits and limitations.

For three common solid propellant grain designs in particular, analytical geometric solutions to surface area over time can be derived using basic geometry, under the assumption that burning propagates uniformly, as well as normal from the interface between propellant and void. The designs and variables for their associated formulas are defined in Figure 2. The derivations solutions yield the following equations of area as a function of burn rate *r* and time *t*:

Tubular:

Rod and Tube:

Cross:

where *Z* is the height of the propellant grain (6.72 m), *R* is the radius of the propellant grain (1.055 m from a 2.11 m diameter), and *r(N)0*terms denote the initial radius of some *N-*th circular geometry used to define the starting limits of voids or propellant. Though the voids in the cross design are bar-shaped, for instance, their widths are assumed equal and can thus be represented by the diameter of a circle starting at the intersectional area between the horizontal and vertical bars.

Even as simple as these designs are, these can begin to push the limits of what can easily be analytically solved. In the case of the cross, for instance, a simplifying assumption was made that burn would progress as though both bars would increase in width until the entire grain is consumed. This does not account for the fact that propellant on the inner corners (being bordered by not one, but two burning surfaces), would burn quicker that along the edges, resulting more realistically in a beveled edge that increases in width from all four corners. This feature can be easily accounted for by adding a term of to the perimetric (bracketed) terms, but for more complicated geometries, accounting for these and similar cases becomes far more difficult.



**FIGURE 3.** Basic concept behind numerical estimation of surface area changes. Objects can be imported or coded where some number (1) represents solid propellant, and a 0-value is given to voids. Propellant cells are subtracted by a desired amount for every void cell they border which has a value below a particular threshold (i.e., 0.3). The simulation proceeds until all propellant values are equal to 0.

The description of this method already makes clear the primary limitations of analytical geometric solutions. That is, solutions are restricted to geometries which not only can be analytically represented in the first place but are also expected to remain somewhat geometrically consistent as burn progresses. On the other hand, if analytical solutions can be found, these will enable rapid solutions and easier calculations.

3.3. NUMERICALLY ESTIMATING PROPELLANT SURFACE AREA

In principle, the grid-based numerical estimation of surface area changes over time is fairly simple, described in Figure 3. For grid-based numerical simulations, the greatest problem for estimating burn profiles comes with attempting to minimize computational intensity. The approaches and improvements to this grid approach will comprise this discussion but is by no means a comprehensive review of other, pre-existing methods.

The grid approach described here populates an array primarily composed of two values: 1 (for unburnt propellant) and 0 (for fully burnt propellant or voids). A negative value (i.e., -999) can be assigned to other cells which will not be considered in the simulation (i.e., the upper and lower corners of the array resulting from a circular cross section. For all voids found on the border of a propellant cell, a desired value is subtracted that is derived from the burn rate of the propellant. Once a specific threshold *T* is reached by the burning cell, surrounding cells are then “ignited” and begin to decrease as well. This process is repeated until all propellant values reduce to 0. The derivation of the subtraction value for each cell is simply computed as follows:

where *M* is the diameter of the propellant grain in pixels and *D* is the diameter of the propellant grain in meters, yielding a value of pixels s-1 and is the fraction by which each cell is reduced. The size of the grid was selected such that *rcell* is considerably less (between 1% and 10%) than the maximum value of any particular cell.

The relationship made using *rcell* only loosely correlates a linear burn rate with the pixel grid and does not account for the impact that different threshold values (*T*) have on burn rate. Figure 4A demonstrates the extent to which the simulated solution still deviates from the analytical solution without accounting for this error. Figure 4B demonstrates that, by scaling the x and the y values of the simulation graph, approximations much closer to the analytical solution are possible. Once these scalar values have been determined for one solution, they are considered as unifying factors that will apply nearly equally to all simulated designs, as described in Figures 4C and 4D.

Additional methods were created to address the occurrence of perturbations in some motor designs (a more in-depth discussion will follow). Three alternative programs and methods were created to attempt to correct this phenomenon, but these anomalies persisted through each. The first method built upon the same algorithm used to detect propellant cells exposed to burning borders, simply counting the size of the array of values returned by the function.

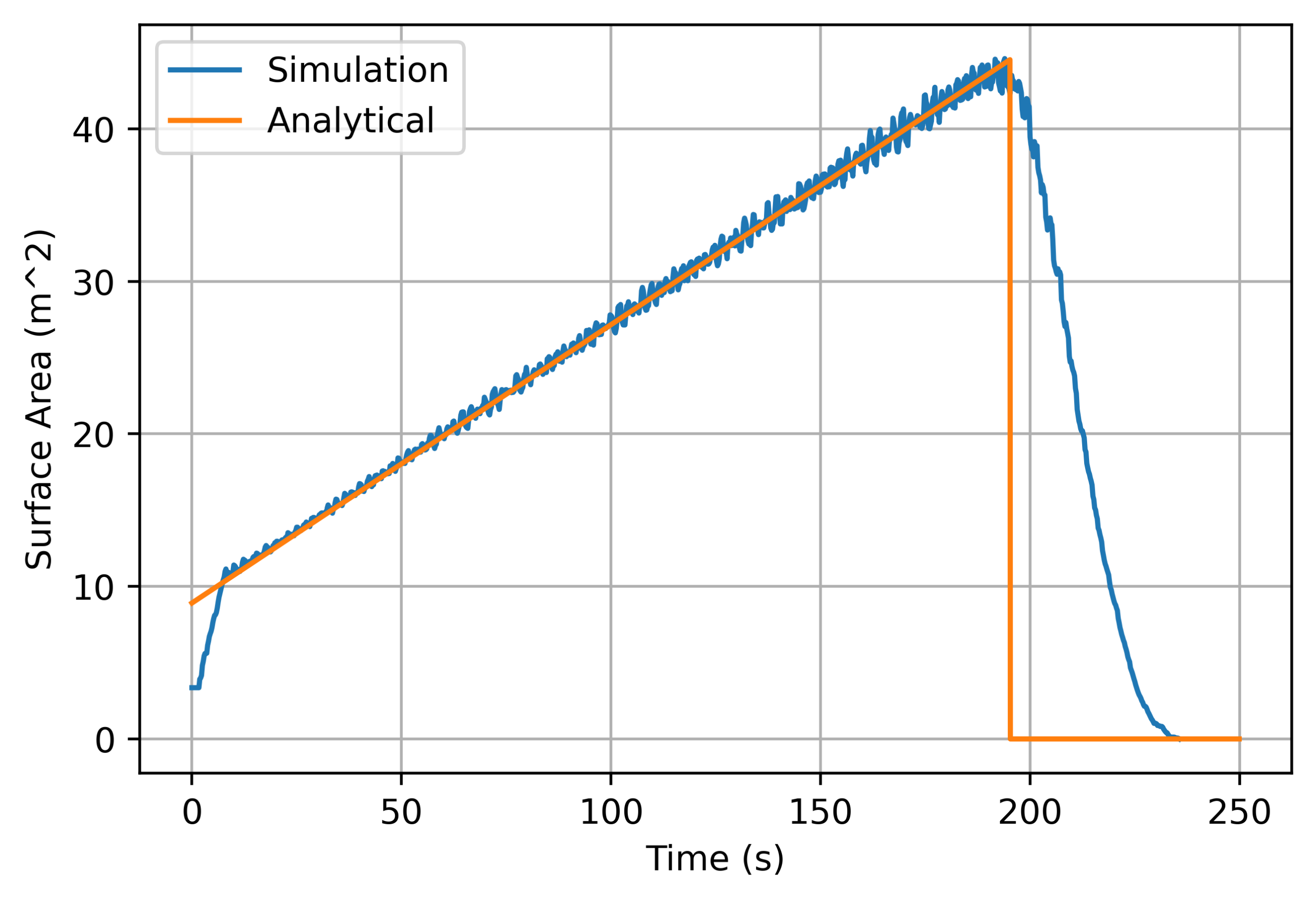
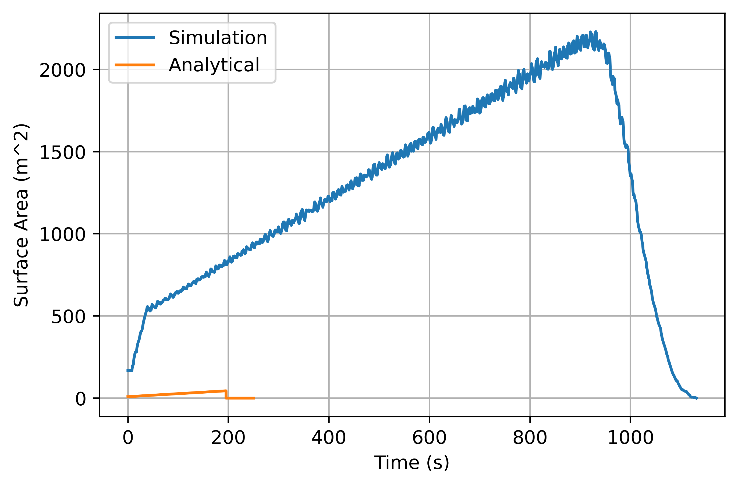
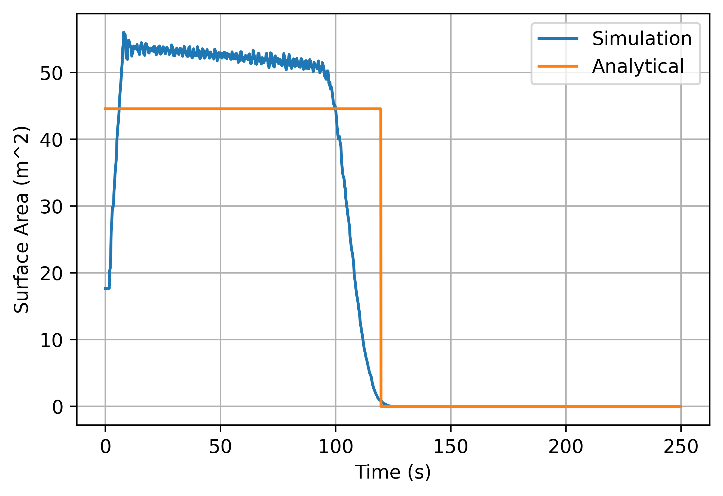
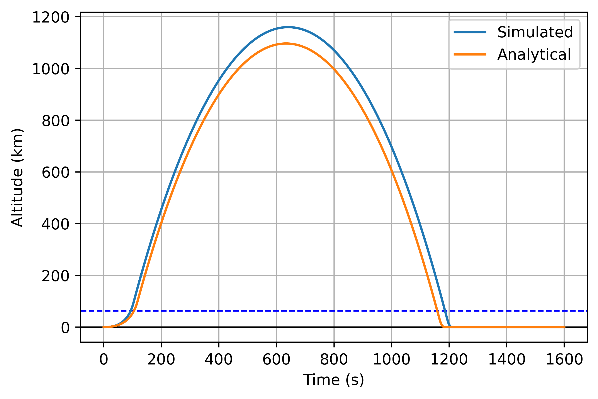
The second method implemented the threshold system already introduced , which added cells to a list when a particular threshold was reached and removed cells when their values zeroed out. Just as was done for the first method, the second method simply counted the size of this array to estimate pixelated surface area. The primary modification made here was that the threshold system counts not only pixels that are directly exposed, but also those who are indirectly exposed via contact to exposed cells. The rationale for this modification was that, with outer cells burning at different rates than those burning incorrectly, this may have led to abrupt drops and rises in exposed cell counts.

The final method attempted to simplify pre-collected data using the previous two methods by using an interpolation function capable of stretching or squeezing arrays to a certain size. By reducing a curve to anywhere between 10 – 400 point, noisy data could be aligned with a more general trend. An error analysis was run, comparing a reduced data set from the cross model to both the raw simulation data and the analytical estimate.

3.4. SIMULATING PROPELLANT BURN

Two methods have been evaluated that follow this procedure in simulating propellant burns. The first method is a tedious approach which iterates through every cell in the array to check its status to determine whether it should be ignited and, if so, how much longer. The second method presumably improves upon the efficiency of this process using NumPy’s *where()* function to identify the location of 0-cells in the map. Using the coordinates from the *where()* function, this method performs fewer checks to iterate through the burn process.

**FIGURE 4**. (A) Unscaled X and Y simulation values deviate significantly from analytical results. (B) Scaled X (0.2) and Y (0.02) values allow for a much better fit. (C) When applied to other models, such as the rod-tube model, these scalars result in loosely fit approximations which can then be fine-tuned. (D) Final trajectory results from the data of 4C illustrate the expectable final deviation from approximations, with maximum 6% error.



**A**

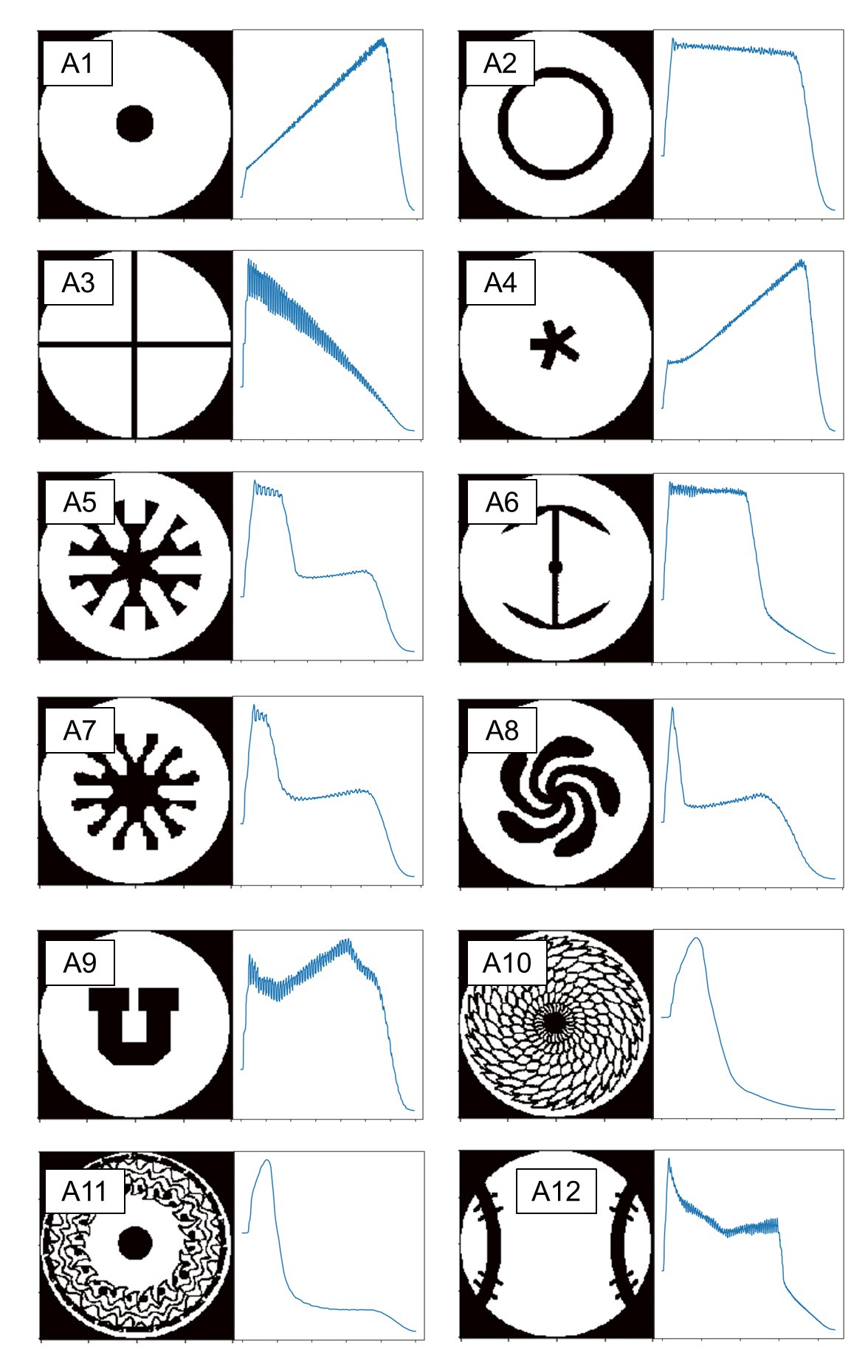
**B**

**C**

**D**

**4. RESULTS**

This section describes the results of the project in the order in which they were found.



BASEBALL

FEATHER

SPIRAL

IRON

STAR

ROD IN TUBE

DREAMCATCHER

UTAH

TUBULAR

CROSS

MULTI II

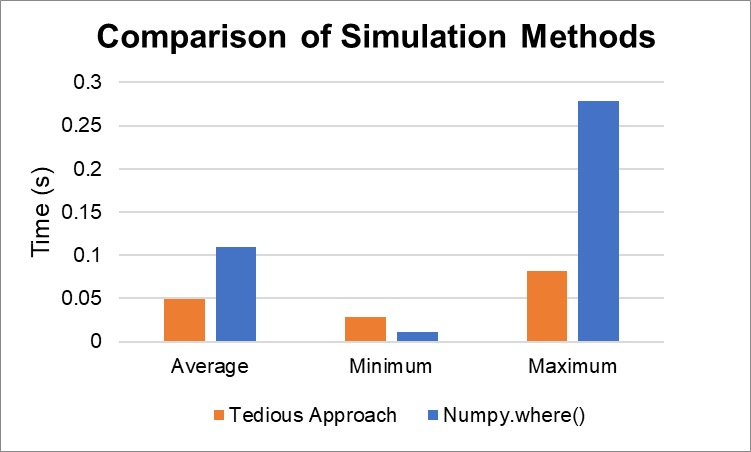
MULTI I

**FIGURE 5**. Plots of 12 different propellant designs to be evaluated in this study, labeled A1-A12 along with nicknames. Designs are shown to the left in each pair, while surface profiles over time are shown on the right

.j’

4.1. DESIGNING PROPELLANT GRAINS

**FIGURE 6**. Comparison of simulation methods demonstrating the generally inferior performance of the where() method, despite managing to achieve a smaller minimum computation time. Obtained using the Tubular (A1) Model.

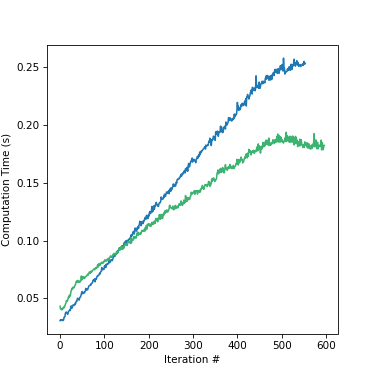


All of the propellant designs to be tested are contained in Figure 5. Significant perturbations were observed throughout A3, A6, A9, and A12. In contrast, more complex models like A10 and A11 yielded very smooth curves. A1 and A4 proved to have extremely similar profiles. A5 and A7, likewise, appear to be analogs, which is not unexpected given that A7 is simply a more organic form of A5. Finally, A10 and A11 also demonstrated highly similar behavior to one another.

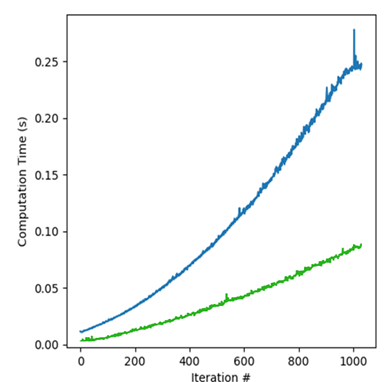
4.2. COMPARISON OF NUMERICAL SIMULATION METHODS

**BURN SIMULATIONS**

The results of performance tests between the tedious approach and the method utilizing NumPy.where() are shown in Figure 6. On average, the tedious approach performed 2.25x better than the where() function, with average values of 50 and 110 milliseconds, respectively. This margin increased to over 3.00x better at the maximum computational times. However, the where() function demonstrated improvements of over 2.6x at the lowest computations, achieving minima around 10 milliseconds compared to the tedious method yielding nearly 30 milliseconds. For both the tedious approach and where() methods, computation times were observed to increase consistently over the course of a simulation (Figure 7).



**FIGURE 7.** Observed computational times for tedious approach (green) and where() approach (blue), demonstrating a significant increase in computation time during the program. Left: Tubular (A1), Right: Rod-in-Tube (A2).



**TRAJECTORY SIMULATIONS**

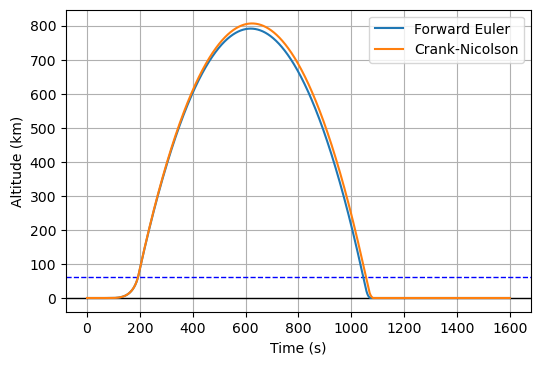
The trajectory simulation was originally run using a time step of 1.0 seconds. This yielded a highly apparent deviation between Forward Euler and Crank-Nicolson trajectories. Using a time step of 0.2 seconds, a maximum deviation of 4.4% (36.0 km) was observed between models. Reducing to a time step of 0.15 seconds proved to significantly increase the time required for computation, yielding a maximum deviation of 3.3% (26.3 km). The deviation between both models expectedly rises dramatically with step size. At a step size of 3.0 seconds, error is nearly 100% between the two. These results are summarized in Figure 8. The results of the trajectory tests themselves are summarized in Figure 9.

**FIGURE 8.** Maximum observed deviations as a function of step size between Forward Euler and Crank-Nicolson models for the analytical A1 grain design.

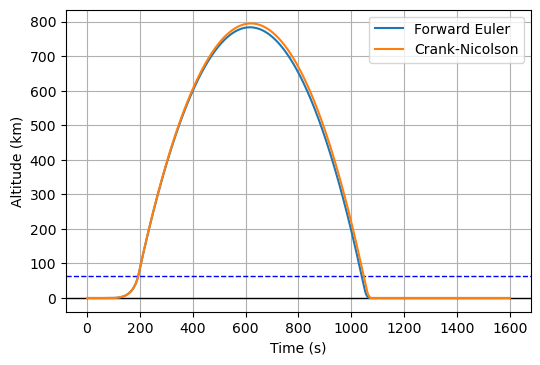
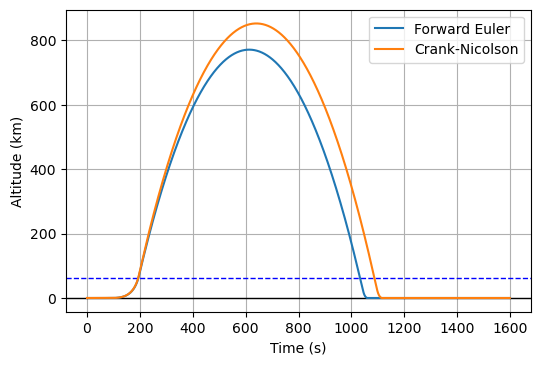


4.3. VALIDATION OF SIMULATION FITS TO ANALYTICAL SOLUTIONS

Simulated models were fit to analytical solutions by multiplying simulation X values by 0.2 and simulation Y values by 0.02. This resulted in a maximum error of nearly 25 km for the tubular trajectory, amounting to about 3.1%. Applying the same scaling to the rod-and-tube model, a maximum error of 110 km was observed, equivalent to a maximum error percentage of 10%. Finally, when applied to the cross model, a maximum error of 60 km was observed, equivalent to an error percentage of 17%. The maximum kilometer error for the cross value results in a greater error percentage than the rod-and-tube model because of the significantly lower maximum altitude achieved by the cross model.



**FIGURE 9**. Performance of Forward Euler and Crank-Nicolson at different time steps. A: 1.00 seconds, B: 0.20 seconds, C: 0.15 seconds

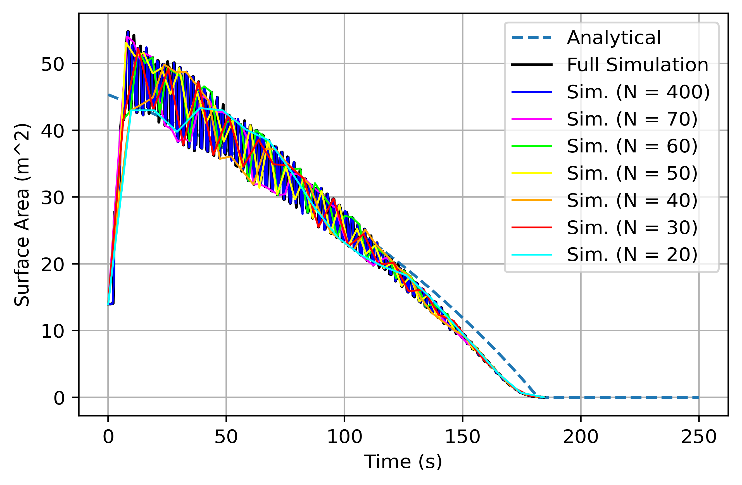


**C**

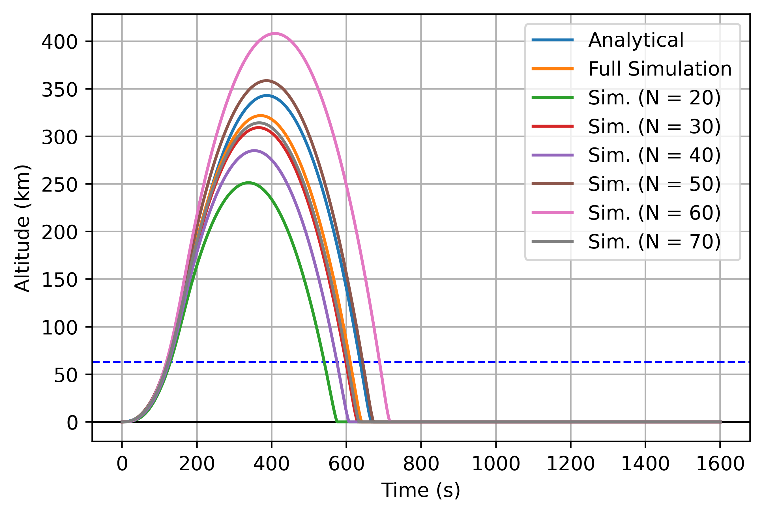
**B**

**A**

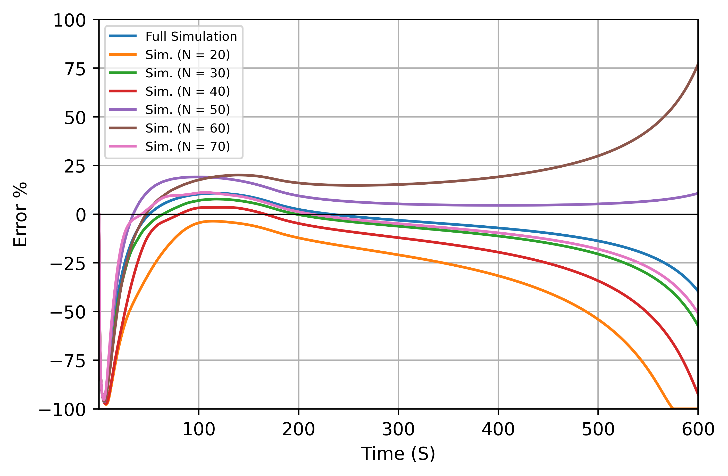
**FIGURE 10.** Reduction of simulation datapoints demonstrates a reduction in fluctuation to attempt a better approximation of the analytical graph



**FIGURE 11.** Trajectory evaluations of cross model (A3) based off of the extent of simulation reduction.



**FIGURE 12.** Errors relative to analytical solution for reduced simulation values for cross model (A3)



Section 3.3 outlined methods for numerically estimating surface areas, including a treatment in which simulated data was reduced from its original number of data points to an arbitrarily lower number, such as that noise in the data could be reduced. The effects these reductions had on trajectory are visible in Figures 10, 11, and 12. The reduction technique was inconsistently effective at achieving lower errors, with the only success being the reduction to 50 points—even so, the full simulation data still performed better for the majority of the simulation. In all other cases, reducing the number of points failed to improve the accuracy of the simulation for a substantial portion of the runtime. This is in spite of the fact that these reductions appeared to converge somewhat onto the analytical solution in Figure 10, though perturbations remained present even in the lowest counts.

4.4. EVALUATION OF PROPELLANT GRAINS ON VERTICAL TRAJECTORY

Validating the analytical solutions opened the possibility of conducting trajectory analyses on each grain design. The data gained from these tests are shared entirely in Figures 13 and 14, including one overall performance chart as well as a set of three different performance tiers to convey individual expected performances more clearly. The tiers are organized approximately by the three analytical models presented, which coincidentally ended up in distinct performance categories.

The low tier, consisting of the fastest-burning, low-altitude configurations, corresponds to the performance of the Analytical Cross (A3) model. Starting with the lowest performers (in terms of altitude), the final ranking for this category is as follows: Feather (A10), Dreamcatcher (A11), Utah (A9), Spiral (A8), Analytical Cross (A3), Multi II (A7), and Simulation Cross (A3). This category had an apogee range between 94 and 412 km, with a flight time range between 300 and 720 seconds.

The mid-tier, consisting of designs with intermediate burn speeds as well as middle-of-the-pack altitudes, corresponds best to the performance of the Analytical Tube (A1) model. Starting with the lowest performers (in terms of altitude), the final ranking for this category is as follows: Multi I (A5), Baseball (A12), Analytical Tube (A1), Simulation Tube (A1), Star (A4). This category had an apogee range between 440 and 803 km, with a flight time range between 300 and 1,100 seconds.

The top tier, consisting of designs with the slowest burn speeds and achieving the highest altitudes, corresponds best to the performance of the Rod-and-Tube (A2) model. Starting with the lowest performers (in terms of altitude), the final ranking for this category is as follows: Analytical Rod-and-Tube (A2), Simulation Rod-and-Tube (A2), and finally, Iron (A7). This category had an apogee range between 1090 and 1200 km, with a flight time range between 1,170 and 1,220 seconds

4.5. EVALUATION OF DIMENSIONAL PARAMETER EFFECTS ON TRAJECTORY

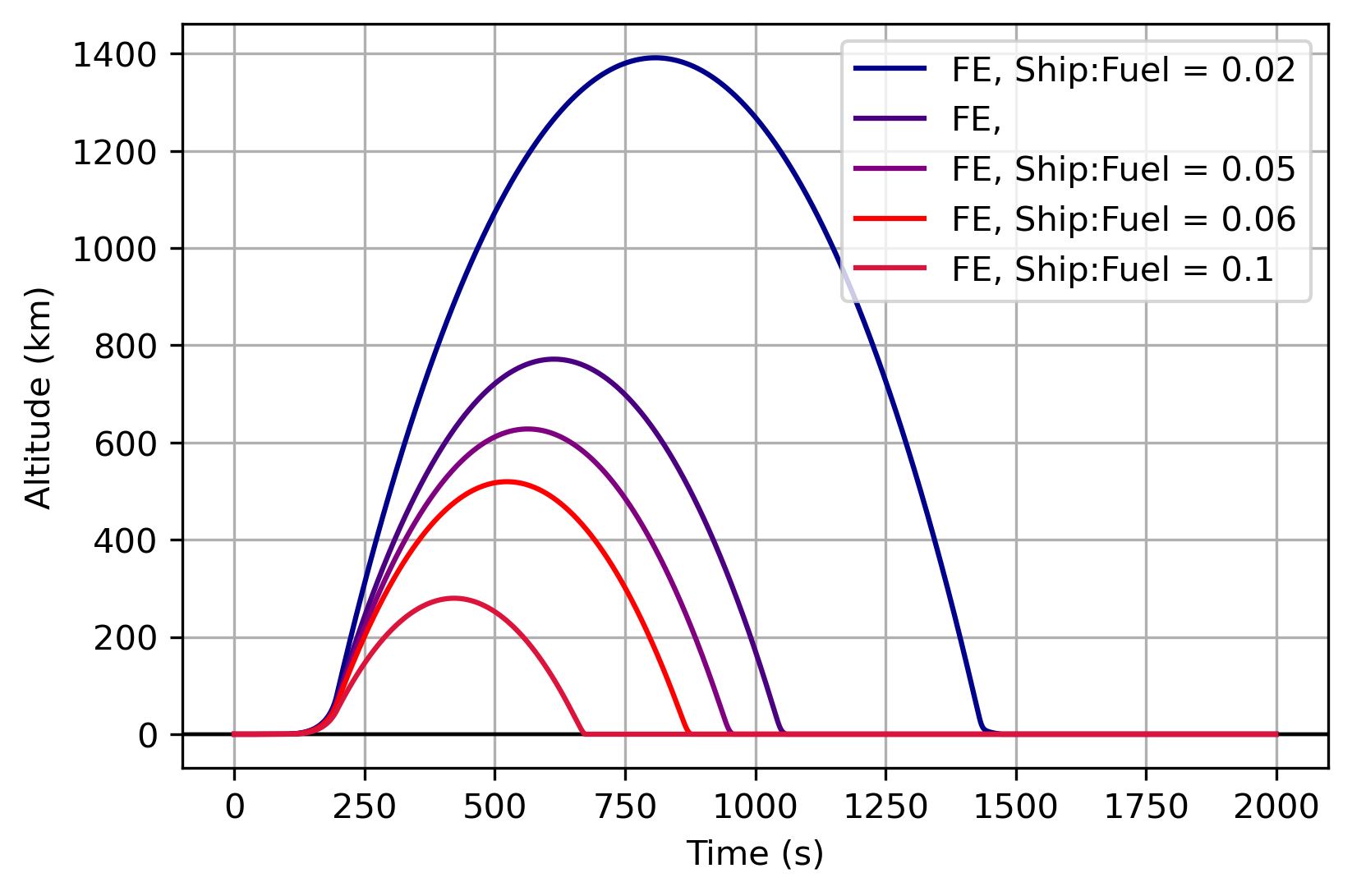
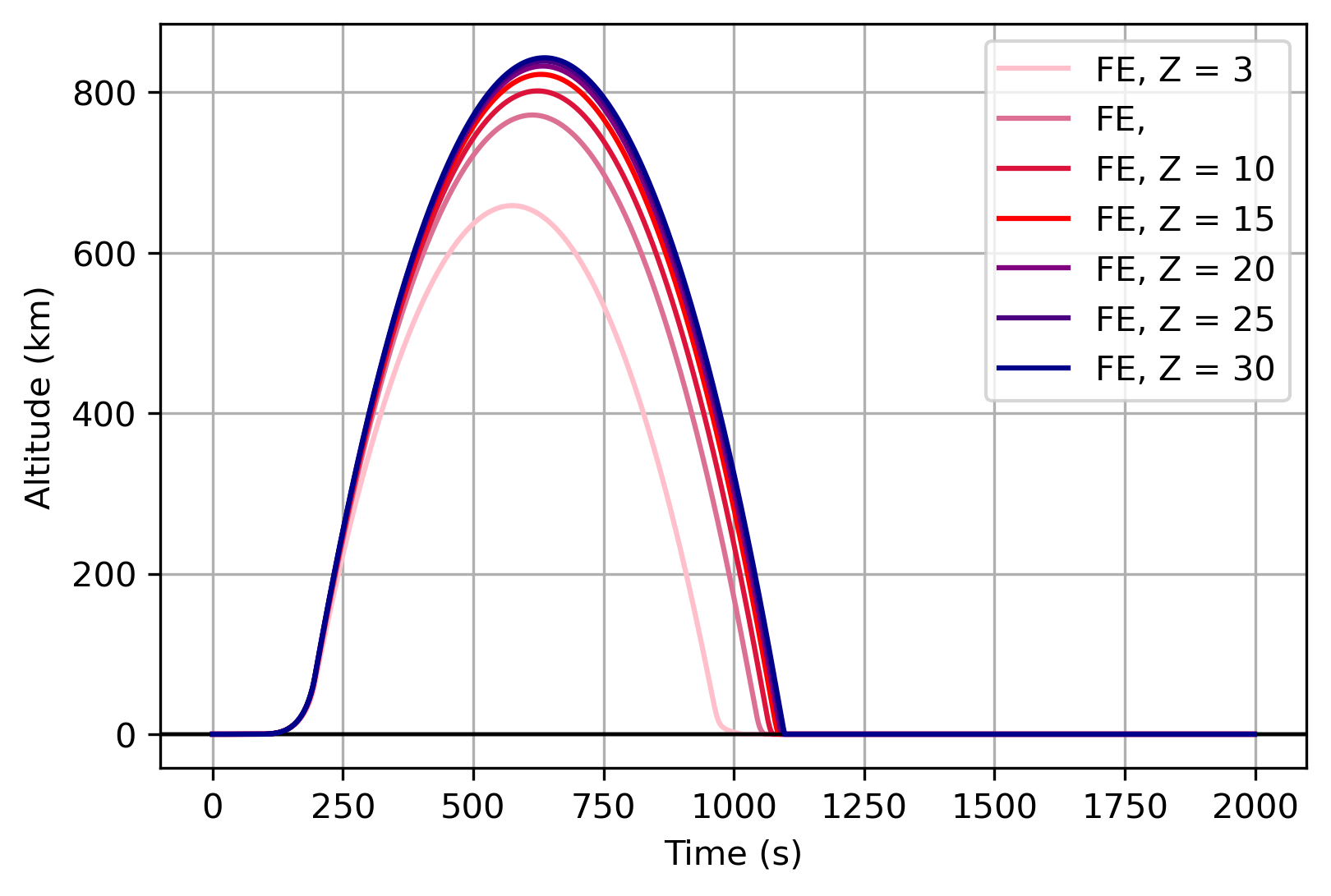
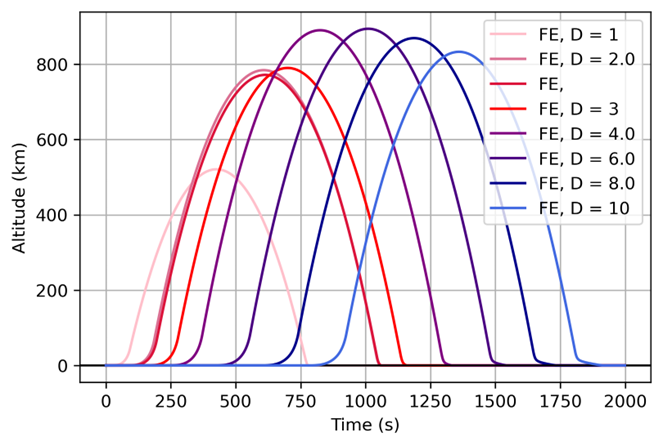
The results from dimensional tests are condensed into Figure 15. In all cases, the tubular model (A1) was assumed. The diameters chosen were done so to compare against the nominal diameter (2.11 m). The first test run, on motor diameter and its impact on flight, resulted in marginal altitude increases while shifting flight time and takeoff significantly ahead (Figure 15A). The lowest diameter tested, 1.0 meter, resulted in a quicker takeoff and significantly lower max altitude than any other diameters tested.

Figure 15B summarizes the impact of different motor height changes. Reducing the height of the motor from nominal (6.72 m) resulted in an immediate decrease in max altitude. On the other hand, even significant height increases, up to nearly a factor of five, resulted in relatively marginal performance boosts.

Figure 15C summarizes the impact of casing mass on flight trajectory from nominal, demonstrating significant impact. The ability to cut down on the nominal fraction (4.05%) proved capable of nearly doubling the max trajectory of the motor. Conversely, increases by a couple of percent in this category resulted in catastrophic downgrades in trajectory altitude.



**FIGURE 13.** Flight trajectories by grain design. Grain designs are noted first by nicknames, then by their corresponding AX number to match the designations in Figure 5.



**FIGURE 15.** Dimensional parameter effects on trajectory. (A) Effects of rocket diameter; (B) Effects of rocket height; (C) Effects of chamber mass as fraction of overall rocket mass at start of flight

**C**

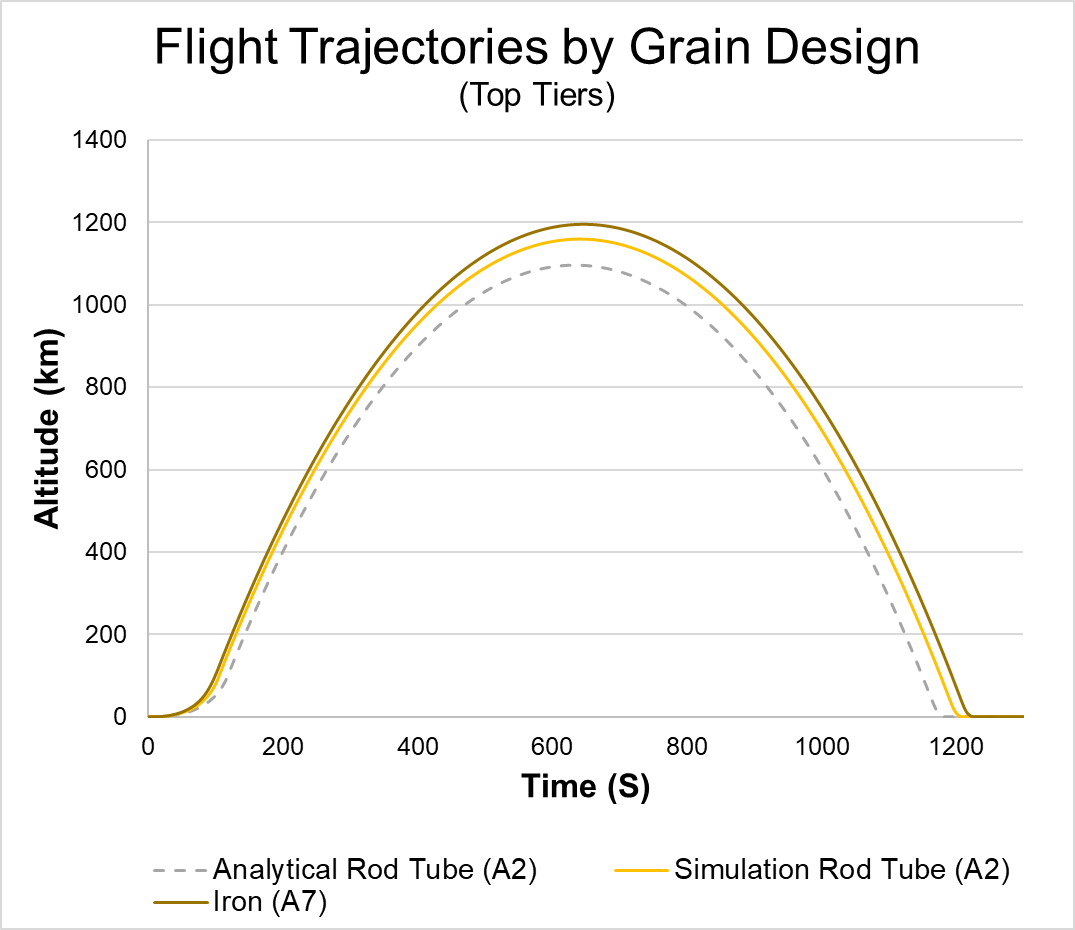
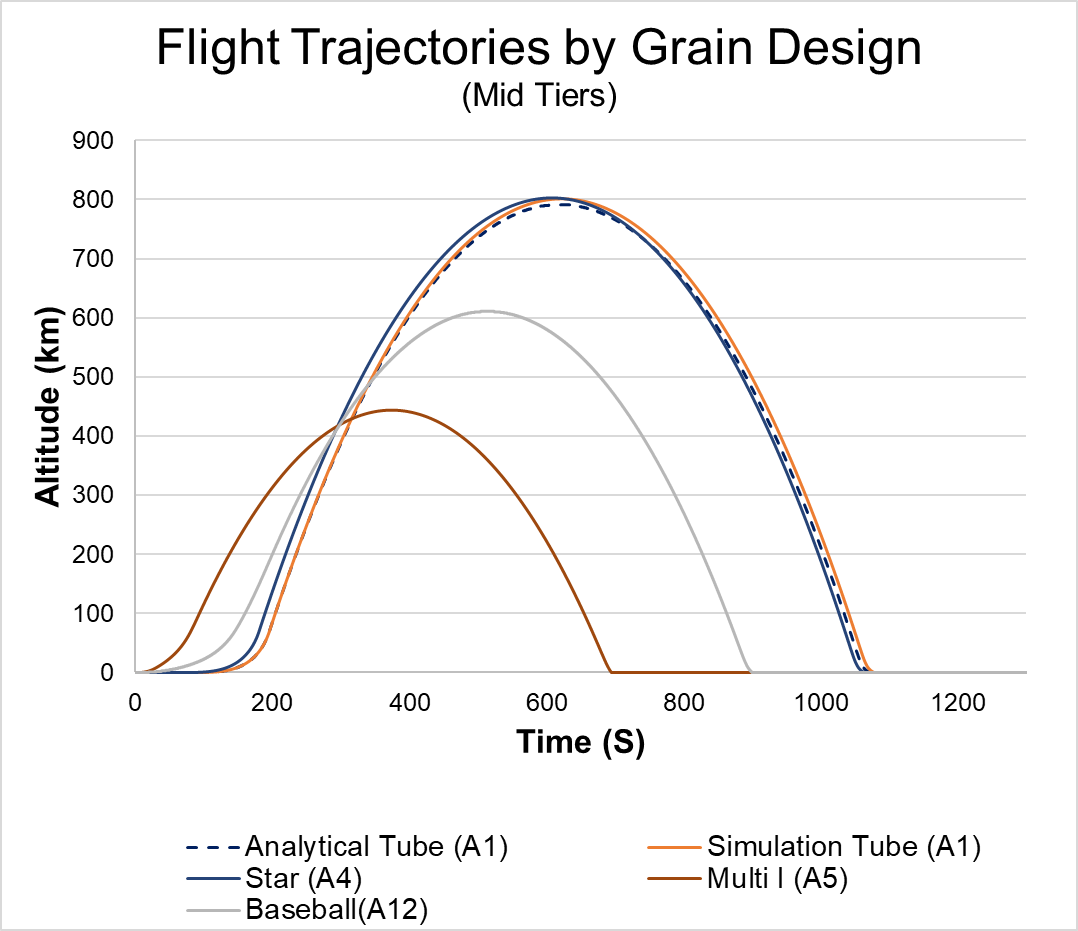
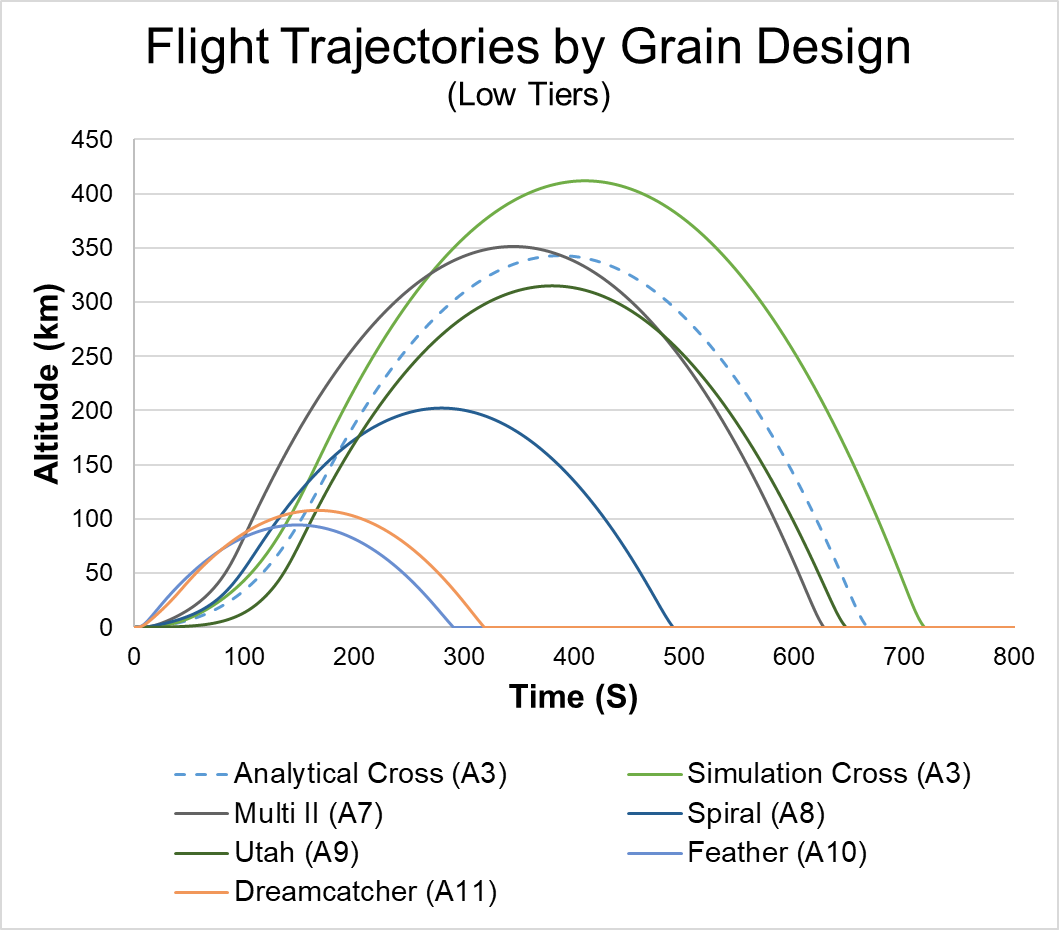
**B**

**A**

Z= 6.72

D= 2.11

Ship:Fuel = 0.04



**FIGURE 14.** Flight trajectories split up into three tiers: Low, Mid, and Top.

4.6 EVALUATION OF STABILITY ANALYSIS ON NUMERICAL METHODS

Typically, an eigenvalue analysis is performed on systems of ordinary differential equations to evaluate their stability. To do this, the derivative of the right hand side of each ODE is taken with respond to each variable solved for and compiled into a Jacobian matrix. This presents a significant issue for the designs presented here, given that the primary focus of this work is to generate and evaluate simulations for propellant burns. Stability analyses may be possible using one of the three presented analytical solutions for area. However, due to time limitations, as well as the primary focus on the development of a propellant burning simulator, this has been deemed beyond the scope of this project.

**5. DISCUSSION OF RESULTS**

This section hosts the discussion of results for each result subsection described in this work.

5.1. DESIGNING PROPELLANT GRAINS

One of the most unexpected aspects of this part of the project was the perturbations that are visible in several of the motor designs in Figure 5. At first glance, these appear to be correlated with increasing surface areas that appear straight and inorganic. The cross configuration, A3, retains its sharp, orthogonal edges throughout the entirety of its burn simulation. Likewise, the U symbol, A9, begins with several sets of vertical and horizontal edges, though these burn into more of a circular shape. This stands in sharp contrast with the most erratic of the twelve designs presented, A10 and A11, which demonstrated no perturbations in their surface area profiles.

At the time of assessing these graphs, it is unclear why or whether a greater number of horizontal or vertical contours in a particular design would contribute to perturbations in the surface area estimations. A variety of alternative counting methods were attempted to remove these effects, but they remained persistent in each iteration. In all cases, the method of surface area estimation has consisted of running an algorithm to count the size of arrays, scaled to contain all pixels that bordered the void area.

5.2. COMPARISON OF NUMERICAL SIMULATION METHODS

Another equally unexpected outcome from this work has been the comparison between the two primary simulation methods tested (tedious and NumPy.where()). It was fully expected that NumPy.where() would perform significantly better than its counterpart. In both the tubular and rod-in-tube simulations, the tedious method generally outperformed. However, the difference between the two were far more pronounced in the tubular simulation, where the where() function increased at a much faster rate.

The original hypothesis purporting NumPy.where() to be superior overestimated the speed at which NumPy.where() could operate, as well as the length of individual calculations for each cell once eligible cells were located. Though it was originally assumed that the tedious method would experience no slowdown in execution time over the course of the simulation, the fact that this occurred seems to suggest that the computations at each point were sufficient to slow down the simulation considerably.

The difference in rates observed in the tubular model does make intuitive sense upon consideration: Whereas the tedious approach tested every array value independently of time, it had a constant number of checks to perform over time. However, in the case of where(), the number of checks increased with time in the case of A1, in which surface area increased at a nearly linear rate.

As for the rod-in-tube model, the where() function performed roughly the same as it did in the tubular model, the primary difference being that it adopted more of a logarithmic curve compared to its roughly exponential curve with the tubular function. Conversely, the tedious model performed much worse on the rod-in-tube model, even exceeding the computation times of the where() model for the first 100 iterations.

The biggest difference between the rod and tube models is the amount of surface area needing to be calculated. There is between a 25-50% increase in surface area from the start for the rod and tub model due to the inclusion of an interior cylinder. Furthermore, this surface area remains relatively constant with time. In theory, this means a higher upfront computational demand for both models, having to perform the required calculations at a greater number of points and more consistently throughout the simulation. This data suggests that the tedious approach is more susceptible to increased surface areas, in terms of computational slowdowns, than the where() model, which remains relatively unchanged.

The fact that the rod and tube model has a constant surface area raises more questions about the behavior of both methods when evaluating the rod and tube model. Why both methods continue to increase in computational time at a roughly linear rate when surface area is remaining constant is unclear. The number of calculations and exposed pixels should be constant relative to the tubular model, suggesting that these increases are not solely the result of an increasing exposed pixel count.

Alternative explanations may deal heavily in how Python behaves over the course of an extended simulation and with large amounts of data. As the simulation progresses, the software is using memory to contain all previous results found in the simulation while continuing to perform its individual computations. It is possible that the memory being stored during the program’s operation has a detrimental impact on its ability to perform additional computations.

5.3. VALIDATION OF SIMULATION FITS TO ANALYTICAL SOLUTIONS.

The fitting of simulation data to analytical solutions presents a variety of problems, many of which have already been mentioned over the course of describing methodologies and results. What has become clear from the fitting tests is that the requirements to fit one set of data to its analytical solution appear to differ somewhat from requirements for another solution and dataset. The fit achieving a relatively low error for the solution it was fit to (tubular, A1) demonstrated increasing errors as it was used for other designs, especially for dissimilar designs (i.e., the cross, A3). This was not entirely unexpected, but was considered an acceptable, rough correlation between iterations and time, as well as pixels and distances.

A variety of alternative methods could be used to validate future simulations. A more careful consideration of input parameters for the burn simulation itself would lend itself well to obtaining data with realistic dimensions, not needing to be fit to analytical models. Specifically, the threshold at which cells not directly exposed to voids begin to burn, and the burn-rate derived subtraction values were found to have the largest impact on burn times. These ultimately were abandoned in this case because the simulation model developed was not equipped to store data on the timescales that would be required to run the simulation to completion. To be able to store and analyze the data for a large number of designs in the server operating the simulation, burn rates and timescales were assumed to be greater than in reality. This way, crashes were avoided and a larger number of tests could be run on different designs. The fitting strategy employed in this work was devised to bring the simulation results back into the reality of the developed analytical solutions.

In summary, the validation of simulation fits revealed significant errors in correlating pixel counts to analytical surface areas. This method was selected due primarily to computational limitations, as well as the inherent limitations of the simulation software developed. The observed errors are considered sufficiently low for the purposes of this work, which primarily focuses on estimations between significantly different propellant grains. Obtaining estimations for grain performance at this stage can be satisfied by rougher estimations, but more specialized analyses must necessarily improve upon these methods moving forward.

5.4 EVALUATION OF PROPELLANT GRAINS ON VERTICAL TRAJECTORY.

In direct opposition to the original hypotheses stated, the vertical trajectory tests have demonstrated that slower burning, longer-lasting grain designs consistently outperform the rest in terms of achieving higher total velocities. Designs that simultaneously achieved high surface areas while keeping them sufficiently low as to not prematurely run out of propellant performed the best. The rod in tube model (A2) as well as the iron model (A6) demonstrated superior performance because of their ability to maintain consistently high surface areas over a longer period of time.

Designs that were meant to maximize surface area exposed an obvious limit on this strategy’s effectiveness, namely, the feather (A10) and dreamcatcher (A11) designs. These achieved much higher surface areas at the very start of burning, resulting in low-altitude trajectories that had characteristically rapid launches. While more sustainable models had delay times, even upwards of 200 seconds, before a positive vertical velocity was obtained, these high-area designs achieved a nearly instantaneous launch by comparison.

The star (A4) achieved a marginally superior performance to the tubular (A1) design. In essence, the star was equal to the tubular design, with the addition of five small triangles to the interior void. This alone increased the overall surface area and must have resulted in a slightly modified, faster-burning iteration of the tubular design.

The iron (A6) appears to have been particularly effective, in part to the fact that it maximizes surface area while keeping voids extremely narrow throughout the grain. This allowed for a maximization of propellant storage with a competitive starting surface area. It should be noted, on almost all of the lower-performing rockets, voids were so extensive that a large portion of the starting volume remained completely unused. It is unclear why the cross, in comparison, performed so poorly despite having similarly narrow voids and appearing to utilize volume effectively. The fact that its surface area steadily decreased over time, while that of the iron’s plateaued for most of its burn time, is likely the determining factor here.

The results of these trajectory tests represent the primary achievement of this work, which set out to design and output trajectory estimations as a function of propellant grain design. These results have been instrumental at highlighting the limitations of rocket motor design and the difficulty in achieving better performance given finite space and material properties.

4.5. EVALUATION OF DIMENSIONAL PARAMETER EFFECTS ON TRAJECTORY

The final part of this discussion centers around the considerations gained from the dimensional testing run on the motor. It will begin by discussing the impacts of diameter, followed by height, ship fuel, and finally, all these factors together.

The diameter of the rocket was expected to have a much more dramatic impact on altitude than in reality. At sufficiently low diameters, catastrophic drops in maximum altitude were observed, as would also be expected at diameters exceeding 10 m. However, the range of diameters between 2.0 and 8.0 m appear to vary to a minor extent, increasing at most by about 100 km. The limitations imposed by diameter are intuitively clear upon reviewing the data. At lower diameters, one approaches the scale of a model rocket, which will lack the fuel capacity required to achieve substantial altitudes. Conversely, at higher diameters, drag experienced by the frontal area of the rocket will increase substantially. Even as burns progress after atmospheric density decreases to 1%, these impacts become apparent once diameters increase beyond 4.0 meters. It seems, based off of this data, that the additional fuel provided by extra diameters is insufficient to overcome gravity and drag at these points.

The biggest impact that diameter appeared to have on the rocket was in delaying the point at which takeoff could occur, demonstrating delays up to 900 seconds. This draws the design of the rocket into question. These delays likely result from excess initial weight preventing the rocket from taking off until a substantial amount of fuel has been burnt. In the most extreme cases, the rocket motor would ignite and burn fuel at ground level for well over ten minutes, demonstrating a substantial waste of resources. In considering this data for future designs, propellant grains need to be adequately designed to produce sufficient thrust at the start of flight to achieve an early liftoff. Were this to occur, we could potentially see a far greater increase in max altitudes as fuel usage is optimized for the size of the motor. Performing this design task, however, is beyond the scope of this paper in particular.

The height of the rocket played a similarly insignificant role in boosting the max altitude of the rocket. Substantial increases in rocket heights, even up to 30 meters, accounted for no more than 100 km extra in the trajectory. Similar to what was observed with small diameters, however, a significant decrease in height was observed to have a substantial impact in lowering the maximum altitude reached, from 800 km to 750 km for a ~50% reduction in rocket height.

It is peculiar at first to note that increasing the height of the rocket does not have a negative impact on takeoff time in any case. This is likely due to the fact that exposed propellant surface area increases at the same rate, resulting in a consistent takeoff time. This was not the case for diameter variations, in which the exposed grain did not scale with radius.

The ship-to-fuel mass ratio appears to have the most dramatic impact on trajectory. For this parameter, a decrease in case mass by half (from 0.04 to 0.02) corresponds roughly to a 75% increase in max altitude, and an extension of flight time by about 375 seconds. However, this mass ratio would likely be the most difficult of these three parameters to obtain in reality, as there are obvious limitations in terms of which materials can feasibly be used to support rocket flight.

**6. SUMMARY AND CONCLUSION**

This work has presented various numerical methods for the estimation of propellant burn and trajectory as a function of propellant burn and other key dimensional factors. In doing so, it has successfully estimated rocket performance for the twelve grain designs presented, in addition to 3 analytical solutions. This was followed by an assessment of impact of three key dimensional variables for rockets: diameter, height, and case mass as a fraction of total starting rocket mass. In so doing, sufficient data has been collected to make statements and evaluations regarding the original hypotheses set forth here.

**What is the impact of different grain geometries on the ultimate performance of a rocket, and how can this be modelled?**

The numerical methods presented have generated insights into how grain geometry affects rocket performance. This model demonstrated a positive correlation between rocket designs which are able to sustain a consistent or increasing grain surface area over the course of a flight and the ultimate success of that rocket in terms of maximum trajectory.

Furthermore, the methods presented highlighted that rapidly burning grain designs perform very poorly in terms of sustained, high-altitude flight. Rockets designed to maximize thrust early on via rapid burns and surface area maximization would seem to lend themselves better to short-distanced missions requiring rapid takeoff.

The methods used have affirmed the effectiveness and validity of two dominant grain designs in achieving sustained, high-altitude flight: The rod and tube (A2) and iron (A6) models. The simulation methods have demonstrated that these designs excel in sustainable flight while maintaining near-constant surface areas. These results, in tandem with those of the other designs, suggest a fine line between excessive and insufficient surface areas that lead to premature burnout or failure to take off, respectively.

**What is the “sweet spot” for a rocket-to-fuel mass ratio that allows for sufficient performance while leaving room for non-propulsive mass?**

This is a difficult question to answer, even considering the data successfully gathered on this question. The results of the dimensional-dependent trajectory tests in Figure 15 demonstrate that minimization of rocket non-propulsive mass is the most influential factor in increasing or decreasing trajectory. As such, it is difficult to make a definitive statement about what weights or weight fractions perform best. The particular rocket-to-propellant mass ratio selection is mission-dependent and should be selected with range, altitude, and overall mission objectives in mind. For maximum-range rocket and missile applications, efforts must be made to minimize non-propulsive mass, such that the highest altitudes and ranges can be achieved.

**At what point (if ever), if propellant is scaled up equally with overall rocket mass, does size begin to have an overall detrimental impact on rocket motor performance?**

Per Figure 15, there is an increase in performance following the increase in diameter to about 4.00 m. The increase in diameter here represents a scale-up of propellant along with overall rocket mass. With this increase resulting in an overall higher max altitude, there is reason to believe that further increases to the diameter have a net negative impact on rocket performance. Once reaching a diameter of 4.00 m, altitude boosts do not continue and begin, in contrast, to lower as drag forces overcome additional propulsive capacity. Note, however, that these results were obtained for a particular fuel geometry and type. There are a variety of other modifications one could make to the rocket which could almost certainly improve the performance of the rocket beyond what has been observed here. These changes would enable the capacity, potentially, to reach much higher altitudes with even larger diameters.

**7. ACKNOWLEDGEMENTS**

* Dr. James Sutherland, who provided insight into the burn simulation algorithm, including the idea to incorporate a threshold system to obtain smoother area curves.
* Richard Nakka, who founded <https://www.nakka-rocketry.net/>, which has been a profoundly impactful and practical resource for estimating key terms and learning the physics of rocket flight.
* Stefan B, user 12998205, on Stack Overflow, who shared a wrapper function *interp1d* which was instrumental in the integration of simulated surface area data with the ODE presented in this issue.
* PROPEP, a software developed by Larry Williams and John Cunningham of Martin Marietta, which allowed for the calculation of key chemical terms and species relationships when provided with default rocket chamber values such as temperature, pressure, and grain composition.

**7. REFERENCES**

Baars, W. J., Tinney, C. E., Ruf, J. H., Brown, A. M., & McDaniels, D. M. (2012). Wall pressure unsteadiness and side loads in overexpanded rocket nozzles. *AIAA journal, 50*, 61–73.

DeLuca, L. T., Galfetti, L., Colombo, G., Maggi, F., Bandera, A., Babuk, V. A., & Sinditskii, V. P. (2010). Microstructure effects in aluminized solid rocket propellants. *Journal of propulsion and power, 26*, 724–732.

Evans, B. (2010). *Nozzle erosion characterization and minimization for high-pressure rocket motor applications.* The Pennsylvania State University.

Refuto, G. J. (2011). *Evolution of the US Sea-Based Nuclear Missile Deterrent: Warfighting Capabilities.* Xlibris Corporation.

Richard Nakka’s Experimental Rocketry Site. (n.d.). https://www.nakka-rocketry.net/

Sutton, G. P., & Biblarz, O. (2016). *Rocket propulsion elements.* John Wiley & Sons.

Tanaka, M., & Shibasawa, G. (2015). Active Interruption of Solid-Propellant Combustion in a Choked Chamber. *51st AIAA/SAE/ASEE Joint Propulsion Conference*, (p. 4177).

**APPENDIX 1. APPENDIX DIRECTORY**

Extensive background research and calculations were performed as part of this project, which have not been discussed in detail in order to maintain a focus on the numerical methods developed and tested. This section will direct attention towards attached files which, collectively, serve as the basis for everything here.

**FOLDER: ANIMATIONS (ALL NOTEBOOKS):** Contains 12 notebooks, each corresponding to a propellant grain design A1-12. These had to be stored individually because Jupyter would crash upon combining more than 3 or 4 in one file. These files allow you to run burn simulations, view their animation, and evaluate computational cost (in time) of the methods used.

**FOLDER: PROJECT FILE NOTEBOOKS:** Contains 4 notebooks that were central in the conduction of this project.

* 1. **TRAJECTORY ANALYSIS.ipynb:**
* Defines and explains background for rocket variables, synthesizing from research and computations included in the appendix.
* Defines functions for gravity, density
* Defines analytical solution functions to A1, A2, and A3 propellant configurations
* Conducts preliminary trajectory analysis using Forward Euler and Crank Nicolson approaches
* Performs error analysis based on time steps
  1. **TRAJECTORY TESTS.ipynb:**
* Imports key variables and functions from the Trajectory Analysis file
* Runs validation of analytical tubular model to simulation, allowing the user to set scales for time and surface area to obtain better fits
* Uses user input on fits to run a trajectory test
* **CUSTOM ZONE:** Gets input from user for grain type and allows for an individual test to be conducted to evaluate a rocket’s performance for a particular grain
* Uses fits to validate A1, A2, and A3 grain configurations
* **REDUCTION**: Evaluates the reduction method to see if reducing the number of simulation data points can be used to eliminate or reduce noise and achieve a better trajectory fit
  1. **MASS AND DIMENSIONAL TESTING.ipynb:**
* Imports key variables and functions from the Trajectory Analysis file
* Repeat tests using Forward Euler to determine impact of diameter, height, and fuel mass composition on performance
  1. **DESIGNING PROPELLANT GRAINS.ipynb:**
* Defines functions for obtaining NumPy arrays for various propellant geometries

**FOLDER: PROJECT FILE PDFS:** Contains the same 4 notebooks as described above in PDF format.

**FILE: PROPELLANT ANALYSIS.XLSM:** Macro Excel document used to compute K values and exit velocity for assumed propellant used in this study. This method is derived from Richard Nakka’s calculation of k for a sorbitol-based propellant.

**FILE: PROPELLANT DATA OUTPUT – NEPE.png:** Summary from PROPEP program of propellant exit stream components, hand-in-hand with the above Excel file.